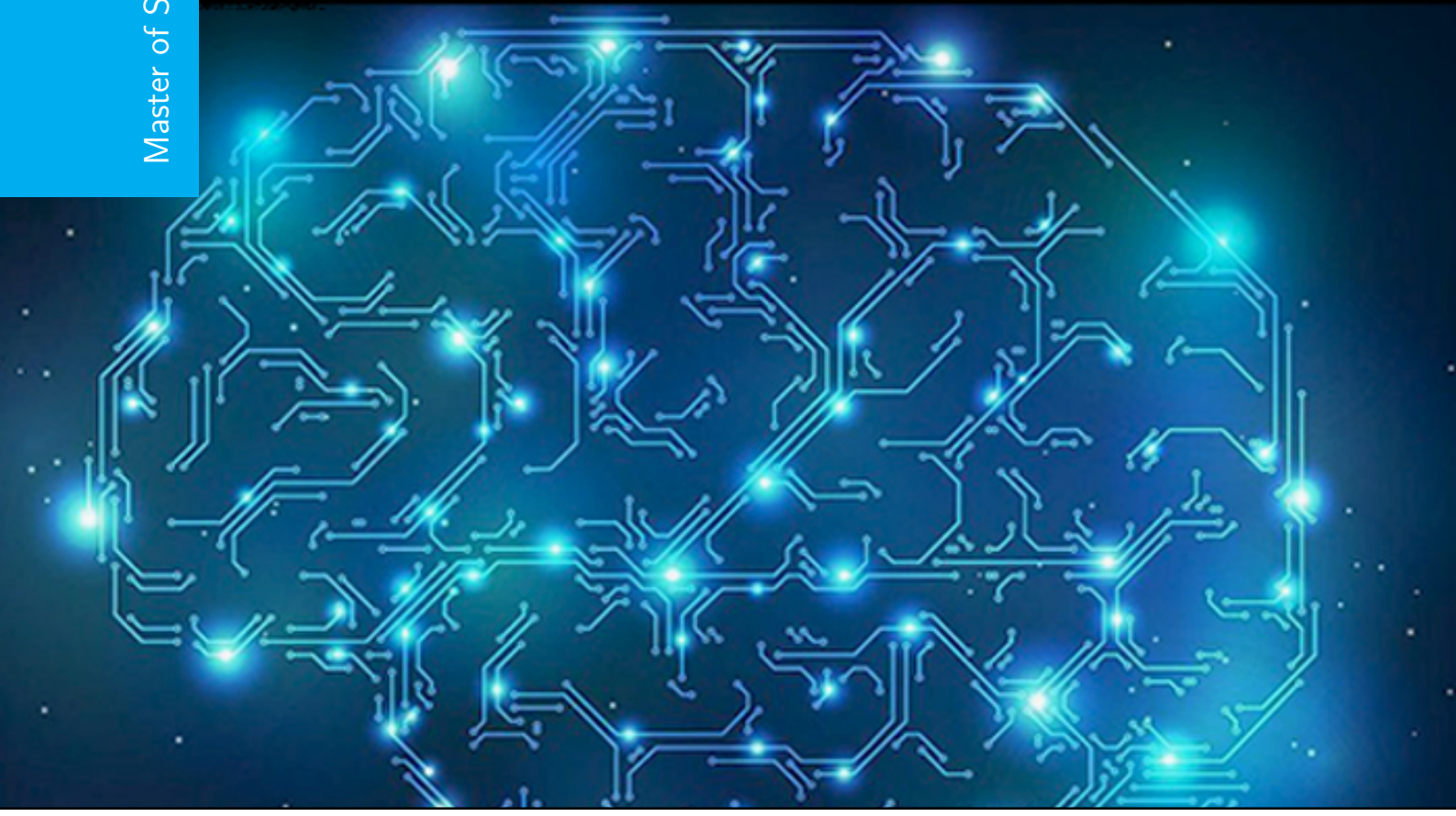
Using Artificial Intelligence for turbulent combustion modelling: Simplifying the conventional lookup tables

Cristopher Morales Ubal

Master of Science Thesis



TU Delft Delft University of Technology

Process and Energy Department

Using Artificial Intelligence for turbulent combustion modelling: Simplifying the conventional lookup tables

by

Cristopher Morales Ubal

to obtain the degree of
Master of Science in Mechanical Engineering
at the Delft University of Technology,
to be defended publicly on Thursday August 26, 2021 at 09:00 AM

Student Number:	4892356	
Thesis Committee:	Prof.Dr.Dirk Roekaerts	TU Delft, Supervisor
	Dr. Ivan Langella	TU Delft
	Dr. ir. M. J. Tummers	TU Delft

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

In reacting flows, detailed chemistry computations are usually avoided precomputing the thermochemical quantities as functions of a reduced set of variables such as the Flamelet Generated Manifold (FGM) approach[34]. Although it mitigates the calculations of detailed chemical mechanism, the memory requirement associated to store the lookup table and retrieve the information during numerical simulations is usually large (order of Gigabytes). Thereby, extending the FGM approach in order to include other conditions requires to add other independent variables which will inevitable lead to increase the size of the lookup table. This will generate that Large Eddy Simulations (LES) cannot be performed such as is the case of the Diluted Air FGM (DA-FGM) approach developed by Xu Huang[9], limiting the simulations to Reynolds Average Navier-Stokes (RANS) approach. In this master thesis, the goal is to use Artificial Intelligence (AI)-Machine Learning (ML) techniques in order to reduce substantially the computational cost of storing lookup tables. In order to achieve this, the Artificial Neural Network (ANN) technique is used. First, a 4D FGM lookup table for hydrogen flames is simplified using the aforementioned AI technique. Then, this technique is used to replace a 6D lookup table generated using the DA-FGM approach. The accuracy and stability of the models provided by ANNs is measured by statistical indicators, providing high accurate and stable AI models. Finally, in the middle of this project, unexpected issues regarding the 4D lookup table were encountered, which lead to recreate the 4D lookup table. After studying carefully how the 4D lookup table was created, a new 4D lookup table is generated, providing excellent results and improving the AI models obtained.

Table of Contents

List of Figures

List of Tables

Acknowledgements

First of all, I would like to thank my supervisor Prof.Dr.Dirk Roekaerts for giving me the opportunity to explore the world of Combustion and Machine Learning-Artificial Intelligence during this master thesis project. I have really enjoyed and learnt a lot during this project. But more important it is the vote of confidence giving to me when I asked for a thesis project, proposing me this challenging topic without knowing me before. Thank you so much for your help and great guidance during the progress meetings. Additionally, I would like to thank Dr. Ivan Langella and Dr. ir. M. J. Tummers for their time and effort examining this thesis.

Furthermore, I would like to thank my friends that I made during this master: Fulin, Ivan and Neel. Finally, I would like to thank my parents and family. Primero darle las gracias a mí hermana Valeska por siempre estar apoyándome en las buenas y en las malas. Sin su ayuda no hubiese venido a estudiar a este país. También a mi cuñado Bastiaan que también me ha ayudo más de lo que debería ayudar un esposo al hermano de su esposa. Muchas gracias Bastiaan por dar un voto de confianza cuando pocos lo hacen. A mi hermano gemelo Ariel por ayudarme y estar siempre en las buenas y en las malas conmigo. Finalmente, me gustaría agradecer a mis papas que pese a todas las circunstancias adversas que le puso la vida por delante, siempre dieron lo mejor por sus hijos y haber logrado este grado es gracias a todo su esfuerzo que pusieron por nosotros en condiciones que otros no hubiesen podido sobrellevar.

Delft, University of Technology
August 26, 2021

Cristopher Morales Ubal

Chapter 1

Introduction

Currently, we are experiencing huge advances in science which have allowed us to solve and simulate numerical problems that years ago would have been impossible to perform. Unfortunately, in some areas such as turbulent combustion there are still issues that prohibit us to perform numerical simulations with high accuracy. One of these issues is the chemical representation provided by lookup tables. Although they allow to avoid integrating chemical mechanisms to numerical simulations, the memory requirement of them (order of Gigabytes) can still be a bottleneck. The present master thesis project will try to address this issue using Artificial Intelligence (AI) in order to replace the traditional lookup tables.

1.1 Motivation

Nowadays, advances on computer technology have allowed us to perform numerical simulations that would have been impossible to achieve in the past decades. These improvements have allowed solving numerically the Navier-Stokes equations (NS) for some turbulent flows (for some specific geometries) for the whole spectrum of temporal and spatial scales involved in such problems (Direct Numerical simulations (DNS)). Nevertheless, DNS are still limited due to computational resources, and an alternative approach is employed: Large Eddy Simulations (LES) which solves the largest scales and models the smallest scales involved in turbulent flows.
This computational limitation is much stronger when computing numerical predictions of turbulent reactive flows where fuel is involved. This is due to the fact that the chemical mechanisms that depict the interaction between flows (fuel and oxidizer) need to be added. As many chemical reactions and species occur in the combustion process, adding detailed chemistry mechanisms into the simulations are currently prohibited due to computational constraints for simulation in 3D[12],[22]. One way of avoiding the addition of detailed chemistry mechanisms is to use precomputed thermochemical quantities that depend on specific reaction coordinates[3]. These thermochemical variables are computed using different models,

one of them is the Flamelet Generated Manifold (FGM)[34], where it is assumed there is a fixed flame structure (flamelet) and the evolution of a scalar quantity in a turbulent flame can be locally represented by the evolution in a laminar flame. In order to address this, flamelets for premixed and non-premixed laminar flames are computed and stored in a lookup table as a function of a reduced set of independent variables. Although using this approach helps to overcome the problem of adding detailed chemistry mechanisms, the storage requirement of lookup tables is large[12]. This is because the main chemical processes in combustion takes place in a thin area. Thus, a high resolution in composition space and physical space is required to capture these main chemical processes. As a result of this, the memory used by high-resolution lookup tables can be of the order of Gigabytes [12],[3].
Therefore, lookup tables can represent a bottleneck in turbulent combustion simulations and simplifying them would help to reduce computational resources.
Here is where Artificial Intelligence (AI)-Machine Learning (ML) enters to the game: Currently, we are experiencing a *Fourth Industrial Revolution*, which is based on the *Third* one and is powered by technologies such as computers and communications (internet). The third one was mainly based on the replacement of manual skills by machines and, on the current one, we are experiencing the replacement of mental skills by machines[32]. In this way, AI embraces all techniques that aim to automate different processes such as industrial, sales and others. At the center of this *Fourth Industrial revolution* is ML which is a part of AI that aims to *learn from examples* or, in other words, to *learn from experience*. ML (formerly known as *Pattern Recognition*) was mainly applied in classification problems in the past decades. However, Recently the use of ML has been extended to regression problems and to other disciplines thanks to the fact that new open source platforms and programming interfaces made the AI-ML more user-friendly outside the data science field. Additionally, new methodologies and better computer technologies have made the training of ML methods easier and faster. In general, ML tries to address the following question: given a data set, is it possible to find out/recognize underlying structures that the data set might have?
In this master thesis project, we are going to apply a ML technique called Artificial Neural Network (ANN) in order to find high-dimensional mappings that will allow replacing the lookup tables. As it will be further elaborated in chapter 3, this technique provides the main advantage of reducing the memory requirement of storing the look-up tables while, at the same time, learning the underlying structure of them.

1.2 Research Questions

The goal of this master thesis project is to address the following questions:

1. Is it possible to use an AI model to replace the chemistry representation provided by a lookup table with four independent variables?

2. Is it possible to use an AI model to replace the chemistry representation provided by a lookup table with six independent variables?

3. Is it possible to obtain highly accurate and stable models that can effectively replace the lookup tables?

4. Is it possible to substantially reduce the memory requirement of look-up tables with the help of AI models?

In the original plan of this master thesis project, the following question was also intended to be addressed:

5. Is it possible to couple these AI models to a Computational Fluid Dynamics (CFD) solver in order to reduce computational resources and be able to perform 3D LES in turbulent combustion problems?

This, however, was not possible to be answered as in the middle of this research project, unexpected issues regarding the 4D lookup table were discovered. This will be discussed in more detail in chapter 6, section 6.1.1. Hence, there were two options to follow: 1) address the original question without fixing the issues found out in the 4D look-up table or 2) address the issues presented in the 4D look-up table and improve the results obtained using ANN. The second direction was chosen, which in the end provided good results.

1.3 Thesis Outline

This report is structured as follows: the next chapter presents a literature review that summarizes previous research related to this master thesis project, analysing and pointing out what advances have been achieved and what problems still need to be addressed (chapter 2). Subsequently, the AI-ML technique used in this project(ANN) is described in chapter 3. Chapter 4 introduces the FGM approach and the lookup tables considered in this project. Chapter 5 provides the numerical setup and software used in this project. Chapter 6 presents and discusses the results obtained using AI. Finally, chapter 7 contains the findings and conclusions of this research project and proposes further recommendations for future research.

Chapter 2

Literature Review

In this section, we are going to depict how Machine Learning (ML) - Artificial Intelligence (AI) has been applied in the combustion community and what has been demonstrated through its use.

2.1 What has been done by the combustion community using AI

In the past decades, the combustion community has applied ML-AI to different problems. As it is reported in [13], ML-AI techniques have been applied to problems that can be classified in two main areas within the combustion field: Combustion systems and internal combustion engines. Likewise, these problems have been addressed using five ML-AI techniques: Expert systems, Artificial Neural Network (ANN), Genetic algorithms, Fuzzy logic and hybrid systems. From [13], we can observe that the main technique used by researchers is ANN, which accounts for 36 out of 69 applications reviewed.

Within combustion systems applications, one of the first works using ANNs in order to avoid numerical computations of including thermochemical equations in the simulations was developed by Blasco et al.[2]. In this work, the authors proposed to model the time evolution of a chemical system using ANNs. In order to achieve this goal, two ML techniques are used : Self-Organizing Map (SOM) and ANN. The first technique was implemented in order to split the thermochemical space in different areas where points are closer between each other (similar idea to clustering techniques) and, once each area is completely determined, an ANN is trained for the data points (thermochemical quantities) that belong to a specific area. Likewise, they obtained the thermochemical points for training the ANNs using a chemical mechanism of five-step and nine-species for air-methane. These were computed using a partially stirred reactor (PaSR) where a transport equation for the joint probability distributions of the scalars is solved. Although, the approach considered in this work differs from the Flamelet Generated Manifold (FGM) approach for creating the lookup tables adopted in this master thesis project, promising results were reported regarding the CPU reduction and

accurate results achieved by using both ML techniques. Additionally, in this work is pointed out a probably issue that could occur using an ANN which is that an ANN could not be able to capture with high accuracy the chemical processes in thin areas. Therefore, also the idea of SOM seems promising for the current project.

In an approach closer to the one adopted in this master thesis project, Ihme et al.[12] developed *Optimal artificial neural networks*(OANNs) for replacing the traditional chemistry representation(Look-up table). It is claimed in this paper that it was the first time that it was possible to use this alternative chemistry representation (OANN) in LES of turbulent reactive flows[12]. In this work, the *flamelet/progress variable*(FPV) is used in order to build a lookup table. The FPV is based on the steady laminar flamelet equations where the thermochemical quantities are parametrized by a reduced set of 3 independent variables: mean mixture fraction, mixture fraction variance and progress variable. In this way, lookup tables with different sizes (different number of nodes used for the independent variables) were generated and compared their memory requirements with respect to memory requirement of an ANN. They are summarized in the following table 2-1:

Table 2-1: Comparison of memory requirements between lookup tables with different resolutions and OANN. Adapted from [12].

Table size	Memory (MB)
$100 \times 13 \times 100$	7
$200 \times 25 \times 200$	53
$400 \times 50 \times 400$	427
$800 \times 100 \times 800$	3417
OANN	0.1

From table 2-1, we can observe that the memory requirement of the highest resolution look up tables is around 30000 times bigger than the memory requirement of an OANN. Additionally, it is also reported that the time required for retrieving information from the ANN is 3 times bigger that the look up table of size $400 \times 50 \times 400$. Nevertheless, it is also pointed out that in LES the time used in retrieval the thermochemical quantities represents less than 3% of the total computational time[12]. Therefore, we can conclude that the increase of time in retrieval information using an ANN would not cause a substantially increase on the overall computational time spent on LES. Furthermore, they performed LES of the Sidney bluff-body swirl-stabilized methane-hydrogen flame using the OANN and the conventional lookup table. From the computations of the temporal spectra of chemical source term and temperature, it is reported that the temporal spectra obtained using a lookup tables converges to the temporal spectra obtained using an OANN when increasing the table resolution[12]. This can be explained as a consequence of the smooth representation of the thermochemical quantities provided by the OANN. Likewise, they concluded that both chemical representations (Look-up table and OANN) provide results that are in agreement with the experimental data. Whereas, in areas where there were disagreements with the experimental data, they concluded that these differences cannot be attributed to a poor performance of the OANN. Finally, it is also pointed out in this work that taking into account heat loss effects, as the Diluted Air FGM (DA-FGM) table created by Xu Huang includes[9], may improve the numerical predictions in turbulent combustion problems. In summary, from this work, we can

conclude that replacing the conventional 3-dimensional lookup table created using FPV by an ANN shows promising results with respect to accuracy and memory requirements.

Recently, at the *ERCOFTAC* course in December 2019, Heinz Pitsch [24] provided different ML techniques for different applications in combustion modelling such as Bayesian method and polynomial chaos expansion for uncertainty quantification in chemical models, data-driven feature extraction and others. Regarding to the ML technique considered in this master thesis project, he used ANNs for improving the fidelity of turbulent combustion simulations. In this presentation, he explained with more details how ANN works and provided some insight of how to select the number of inputs for training an ANN. Additionally, two ML techniques for *Feature extraction* were given: Principal Component Analysis (PCA) and Kernel PCA. Although, the mathematical explanation of these two techniques are out of the scope of this report, a general idea of them is to map the inputs into a lower dimensional space in order to reduce the input dimensionality. These techniques may be useful in order to improve the *Power of generalization* of the model when the size of the data set is small and the number of inputs is large (this phenomena is called *Curse of Dimensionality*). Notwithstanding, for the current size of the lookup tables considered in this master thesis project (order of millions of points), this issue may not be expected.

At the same course, Cuoci [5] presented a detailed overview of ML models for combustion problems with detailed kinetics. In this lecture, it is shown how ML models can be used for acceleration of chemistry with respect to ordinary differential equation system solutions, local reduction of chemical complexity and reducing the number of detailed chemistry calculations per iteration. For the purpose of the current project, it is explained how ANNs can be applied for reducing memory storage with keeping accuracy and computational cost. In this way, a review is given of the work of Ihme et al.[12] explained previously and two important issues were pointed out that could eventually occur in the implementation of this master thesis project:

- Complex ANN's architectures are required in order to capture more complex chemical behaviours when minor and pollutant species are considered
- A more complex *ANN*'s architecture will imply higher retrieval times with respect to conventional lookup tables. Therefore, the use of ANNs in numerical simulations of multidimensional turbulent reacting flow could increase considerably the computational time.

In relation to this, it is also suggested the idea used by Blasco et al.(SOM technique)[2] of training localized ANNs in order to avoid the issue of slowness that a complex ANN could cause.

Following this idea, Ranade et al.[27] proposed to combine ANNs and SOM to replace a lookup table in order to address the computational efficiency issues that could result from using complex ANNs. In this approach, the lookup table is created using the *Laminar flamelet* approach where the thermochemical quantities (density, temperature, mass fractions and others) are parameterized by four independent variables: mean mixture fraction, mixture fraction variance, mean dissipation rate and total specific enthalpy. In this work is reported that partitioning the thermochemical space in areas (SOM technique) results in reducing more than

half the training time for *ANN*s. Similarly, a reduction of the network complexity is reported, where at each hidden layer fewer neurons need to be considered in order to achieve the same accuracy. Likewise, retrieval times between complex ANNs and ANNs +SOM are compared showing faster results using both ML techniques. Finally, RANS and LES are conducted in a DLR-A turbulent jet flame in Ansys Fluent 19.0. A comparison between retrieval times using linear interpolations from the lookup table and ANN+SOM model is reported showing that retrieval times of both approaches are in the same order (ratio 1:1.2, approximately) opposite to the increase of retrieval times reported by Ihme et al.[12] (3 times higher using OANN than conventional lookup table). Although this is an issue that could eventually occur during the implementation of this master thesis project, it must be pointed out that the activation functions that will be used in this master thesis, Rectified Linear (RELU) and Sigmoid (see section 3.1, expressions (3-1) and (3-3) respectively), are simpler than the Hyperbolic Tangent (see equation (3-4)) used in the work of Ranade et Al[27]. Therefore, their use may be expected not to increase considerably the retrieval time spent using linear interpolations in the conventional lookup tables.

One of the most recent works regarding simplification of the chemistry representation in turbulent combustion problems has been proposed by Zhang et al.[35]. In this work, the authors generated a two-dimensional lookup table using the FGM approach, the two independent variables considered in this work are: mixture fraction and progress variable. Subsequently, the generated lookup table is used in order to train an ANN where the authors adopt the following criterion: Species which mass fraction error is lower than 5% between the prediction obtained by an ANN and the value provided by the lookup table are retrieved in Computational Fluid Dynamics (CFD) simulations using ANNs. Meanwhile species which error larger than 5% are retrieved from the conventional lookup table. Thus, Reynolds Average Navier-Stokes (RANS) and Large Eddy Simulations (LES) with the combined use of FGM table (for some species) and ANNs (for the rest species), which authors have denominated FGM-ANN, are performed in the Engine Combustion Network (ECN) Spray H flames. In this way, high accuracy and efficiency are reported comparing the LES results and the experimental data. Although a memory reduction of more than 8 times is reported with respect to the storage of a complete lookup table, the retrieval time of using FGM-ANN is three times longer than the *FGM* table. This results was also found by Ihme et al.[12].

2.2 Conclusions about previous works

Previous works using ML-AI by the combustion community have been reviewed. From this review, we can conclude the following:

- ML-AI has been reported to have good capability and performance in turbulent combustion problems.
- For the goal of this master thesis, the replacement of conventional lookup tables reported by Ihme et Al[12] shows promising results in terms of accuracy and computational memory savings.
- Nonetheless, it must be carefully checked whether an ANN can effectively capture highly nonlinear process in thin areas which are expected in combustion problems. As it has

been reported in [2], using SOM technique could be useful for tackling this probably issue.

- Although the SOM technique has been reported to be very useful in reducing the retrieval time, it is not straightforward how to partition the thermochemical space for the lookup tables that we aim to simplify in this project. This is because the numerical representation of the manifolds created using the FGM approach are quite structured(points are equidistant between each other in each direction) . Hence, the partition would be mainly based on the outputs. However, this would imply making decisions on how to treat points that would be in the boundaries of the areas obtained by the SOM technique. Thus, while it would reduce the retrieval time, it could cause an additional problem of how to define points in the boundaries.

- As reported by [5],[12], the computational cost for retrieving information from ANNs is more expensive than the usual linear interpolations made by conventional lookup tables. Therefore, this could become an essential impediment in order to effectively replace lookup tables by ANNs. Consequently, it might be studied when the preliminary simulations of this project are completed.

- Likewise, the network architecture must be chosen with care in order to reduce the interpolations overhead and training time as it is mentioned in [5] and [27].

- As in [27] this higher retrieval time is mainly attributed to the activation function used in that work (Hyperbolic tangent), the use of a much simpler activation function such as RELU may alleviate that computational cost.

- Promising results of using an intermediate solution that combines part of the lookup table and ANNs for replacing a two-Dimensional lookup table have been reported in [35]. Although this intermediate approach may be a suitable solution for small lookup tables (2-3 independent variables), for higher dimensional lookup tables (four or more independent variables) the memory requirement of partially store a lookup table may still prohibit numerical simulations of multidimensional turbulent combustion problems.

- Finally, from the previous results of considering 2, 3 and 4 independent variables, it may be expected that replacing four and six-dimensional lookup tables that incorporate additional physical processes (differential diffusion and dilution effects, respectively) by ANNs may provide more accurate results in LES and RANS in multidimensional turbulent combustion problems.

Chapter 3

Artificial Neural Networks

This section, we are going to explain what an Artificial Neural Network (ANN) is and how it works. Finally, advantages of using ANN will be listed. These notes are based on books [15],[1] and [32].

3.1 Main idea

An Artificial Neural Network (ANN) also known as Multilayer Perceptron (MP) is depicted as an attempt to represent the information processing in biological systems in a mathematical way. In Figure 3-1, we can observe a schematic representation of a biological neuron.
A general description of how it works is as follows: *Dendrites* transmit the signals received from input neurons to the cell body of the receiving neuron. Before arriving at the cell body, this information is pre-processed in a process called *Synapsis*, which can be seen as a *weighted connection* [28]. Once the weighted information has arrived to the cell body, this is transformed at the nucleus. After being transformed at the nucleus, the transformed signal is transmitted to other neurons through the *Axon*.

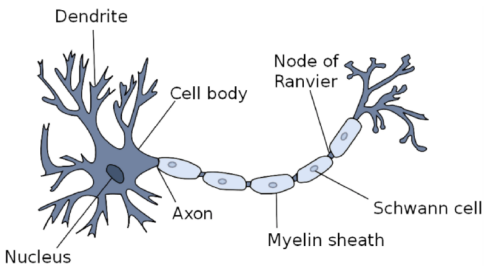


Figure 3-1: Schematic representation of a biological Neuron.
(taken from [28])

This biological mechanism is captured in the *Perceptron concept* which is shown in the following Figure 3-2:

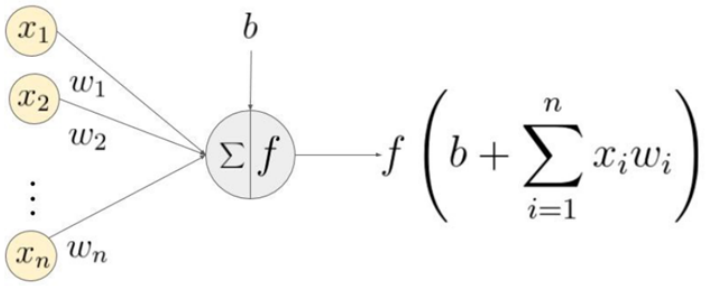


Figure 3-2: Schematic representation of a Perceptron.

From Figure 3-2, we have the following:

- x_i : Inputs (signals/information) transmitted to the neurons.
- w_i : *Weights* or *Adaptive parameters* that mimic the *synapsis* process in the dendrites. They can also be thought as how much important is an input x_i for a neuron.
- b : *Bias* or *Threshold* is a term that shifts the activation function. It is also an *Adaptive parameter*.
- $f(x)$: *Activation function* which is responsible of transforming the weighted information in the neuron.
  In principle, any function can be used as an activation function. However, the choice of the activation function depends on the data and target variables. The most common activation functions, due to their simplicity and for other reasons that are out of the scope of this report (for more details, see chapters 3 and 4 of [15],[33]), are the following:

$$\text{Rectified Linear(RELU)} \ : \ f(x) = \begin{cases} x & , \text{ if } x > 0 \\ 0 & , \text{ if } x \leq 0 \end{cases} \tag{3-1}$$

$$\text{Linear} \ : \ f(x) = x \tag{3-2}$$

$$\text{Sigmoid} \ : \ f(x) = \frac{1}{1 + \exp(-x)} \tag{3-3}$$

$$\text{Hyperbolic Tangent} \ : \ f(x) = \tanh x \tag{3-4}$$

The output of the information transformed by the activation function is denoted as:

$$z = f\left(b + \sum_{i=1}^{n} w_i x_i\right) \tag{3-5}$$

Thus, if we arrange more than one neuron in series and parallel and fully connect them in a layered way (it means that each neuron in any layer is directly connected to every neuron in the previous layer), we create an ANN as it can be seen in the following Figure 3-3:

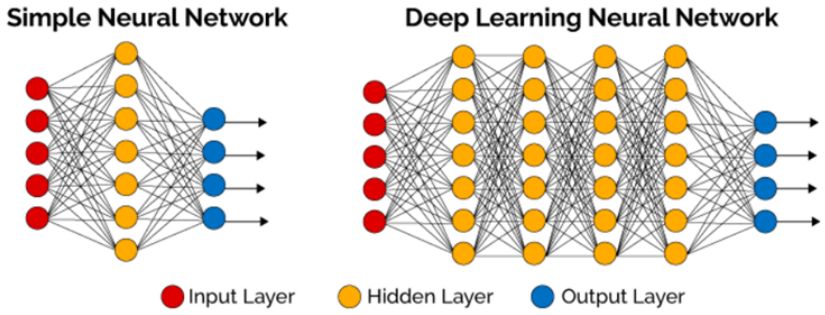


Figure 3-3: Single Neural Network (also known as *Shallow ANN*) and Deep Learning Neural Network (DL)

From Figure 3-3, we have that the first layer is called the *input layer* where inputs feed the network, these inputs are transmitted to the *hidden layers* which are the hidden neurons that transform the inputs (mapping to a higher dimensional space) and the last layer called *Output layer* where the prediction value is computed. At each neuron in the hidden layers, the activation function is evaluated using the signals from the neurons at the previous layer, this can be seen as a *Forward propagation* of the inputs through the network.
In this way, the output of each neuron at each hidden layer in an ANN is given by a composition of functions as follow:

$$\text{Output of k} - \text{th neuron at First hidden layer} : z_k^1 = f\left(b_k^1 + \sum_{i=1}^{n} w_{ik}^1 x_i\right) \tag{3-6}$$

$$\text{Output of k} - \text{th neuron at r} - \text{th hidden layer} : z_k^r = f\left(b_k^r + \sum_{i=1}^{k_{r-1}} w_{ik_{r-1}}^r z_i^{r-1}\right) \tag{3-7}$$

For regression problems, Rectified linear and linear are the most commonly used activation functions in the *hidden* and *output layers*, respectively. Therefore, in this report, Rectified Linear (RELU) will be used. Likewise, Sigmoid will be employed in the *Hidden* and *Output layers*, and it will be shown which one provides the best fitting in terms of accuracy and stability.

3.2 How to define the number of neurons per layer and the number of hidden layers?

In order to define the architecture of our ANN, we can consider the Universal approximation theorem (UAT) that states the following: A single hidden layer can approximate any continuous function from a finite-dimensional space to another with any desired non zero error providing enough hidden neurons[8].
In other words, this theorem states that there exists a mapping where inputs can be transformed, and the regression problem can be successfully solved. Nevertheless, this theorem does not provide what conditions are needed in order to obtain a good approximation (for instance, how much data and how many hidden neurons). Therefore, an ANN with One single hidden layer is not the unique solution to all problems.

Additionally, in order to achieve that the ANN can effectively learn and generalize from a given data set, the number of hidden neurons must grow when the data set grows[7].
On the other hand, there exists a theorem obtained by *Kolmogorov* and adapted to *Neural Networks* that states the following: any continuous mapping from d input variables x_i to an output variable y can be represented exactly using two hidden layers having $d(2d+1)$ neurons in the first hidden layer and $(2d+1)$ neurons in the second hidden layer[1]. Notwithstanding, this theorem does not state the activation function needed in order to achieve that perfect representation. Additionally, it cannot be expected to represent with a fixed number of activation functions an output function $y(\mathbf{x})$ which has infinite degrees of freedom[1].
Therefore, acknowledging the previous considerations, the choice of the number of layers and neurons per layer will be made using an iterative procedure . This procedure will be explained in the chapter 5.

3.3 Training an ANN

We aim to obtain prediction values that can be very close to the real values. Thus, in order to achieve it, the goal is to minimize the *sum-of-squares error* $E(\mathbf{w})$ (also called *Loss Function*) given by:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n} \left\|\widehat{\mathbf{y}}(\mathbf{x}_i, \mathbf{w}) - \mathbf{y}(\mathbf{x}_i)\right\|^2 \tag{3-8}$$

Where:

- $\widehat{\mathbf{y}}(\mathbf{x}_i, \mathbf{w})$ is the predicted value computed for an input $\mathbf{x}_i$ by the ANN.
- $\mathbf{y}(\mathbf{x}_i)$ is the real value for an input x_i.
- $\mathbf{w}$ the adaptive parameters that are adjusted in order to minimize E. For simplicity, the bias term b is included in $\mathbf{w}$.

Thus, given a training set $\{(\mathbf{x}_i, \mathbf{y}_i),\ ,\ i = 1, \ldots, n\}$, we train our ANN with this data set in order to minimize E. Note that inputs $\mathbf{x}_i$ and outputs $\mathbf{y}_i$ can be vectors of any dimension. We will come back to this point in the final chapter of this report.

3.4 Optimization of adaptive parameters

As we are trying to minimize the loss function (3-8), the goal is to find $\mathbf{w}$ such as $E(\mathbf{w})$ is the smallest value possible. It means where its gradient vanishes:

$$\nabla E(\mathbf{w}) = 0 \tag{3-9}$$

We can get closer to that minimum moving in the direction of $-\nabla E(\mathbf{w})$ which allows reducing the *Loss function*. We can achieve (3-9) using an iterative numerical scheme where the adaptive parameters $\mathbf{w}$ are updated at each iteration using *the gradient information* as follows:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau} - \mu \nabla E(\mathbf{w}^{\tau}) \tag{3-10}$$

Where μ is called *learning rate* and can be thought as the rate of how the ANN learns a specific task. Additionally, it can be thought as a way of keeping the neural network alerts to sudden changes while it is being trained. The numerical scheme (3-10) is known as Gradient Descent (GD).
Other variations of the GD scheme exist, such as Nesterov's Momentum, AdaGrad, RMSProp, Adam and others. However, discussing these numerical schemes is out of the scope of this report. For more details see [32], chapter 18.

3.5 Error back propagation

In (3-10) a gradient of the error needs to be evaluated at each step. This is done by sending back through the network the error at the output layer. This process is known as Error Back Propagation (EBP). From (3-8), we aim to compute the derivative of E at the output layer for a specific input $\mathbf{x}_n$ with respect to the weights of the k-th neuron at the r-th layer $\mathbf{w}_k^r$. If we previously define the argument of the k-th neuron at the r-th layer as follows:

$$\theta_k^r = b_k^r + \sum_{i=1}^{k_{r-1}} w_{ik_{r-1}}^r z_i^{r-1} \tag{3-11}$$

We can prove using the chain rule that:

$$\frac{\partial E}{\partial \mathbf{w}_k^r} = \underbrace{\frac{\partial E}{\partial \theta_k^r}}_{=\delta_k^r} \frac{\partial \theta_k^r}{\mathbf{w}_k^r} = \delta_k^r \mathbf{z}^{r-1} \tag{3-12}$$

where $\mathbf{z}^{r-1} = (1, z_1^{r-1}, \ldots, z_{k_{r-1}}^r)$ is a vector of the outputs of the neurons at the r-th layer. Note that 1 is included in the first component because it is related to the bias b_k^r. The δ_k^r's are known as *errors* for the reason that will be explained as follows. The next task is to compute δ_k^r, for the last output layer, $r = L$, it can be proven that:

$$\delta_k^L = \underbrace{(\widehat{y}_k - y_k)}_{e_k} f'(\theta_k^L) \tag{3-13}$$

where e_k is the error associated with the k-th output predicted by the ANN, this is the reason of the name given before. For the hidden layers, using the chain rule, the δ_k^r's can be computed recursively using the following expression:

$$\delta_k^{r-1} = \left(\sum_{i=1}^{k_r} \delta_k^r \mathbf{w}_{ik} \right) f'\left(\theta_k^{r-1}\right) \;\; ; \; k = 1, 2, \ldots, k_{r-1} \tag{3-14}$$

In summary, the training procedure can be depicted as follows:

1. Randomly initialize the adaptive parameters w_{ik}^r at each neuron in the network.
2. Feed the network with the training inputs $\mathbf{x}_i$ and *forward propagate* them through the network computing z_k^r at each neuron using the activation function (3-5) and equations (3-6)-(3-7).

3. Compute the errors δ_k^L's at the output layer between the predicted value and the real value (3-13).

4. *Backpropagate* them through the network using (3-13)-(3-14) and update the *adaptive parameters* using (3-12) and (3-10).

Once this process of *feed-forward* and *backpropagation* has been applied to the whole training set, it is said that an *Epoch* has been completed. In general, more than one *epoch* needs to be completed in order to obtain the desired error and accurate predictions.

3.6 Preventing E to get stuck in local optima

The process of updating the adaptive parameters can be done after the neural network has been fed with the whole training set and computing (3-12) as the *mean value of the errors* after one *epoch* has been completed. This approach is called *Batch or offline GD*.
Nevertheless, as E in (3-8) can be highly non-linear on $\mathbf{w}$, it can have many *local minima and/or saddle* where we will have (3-9), but we will not have found a *global minimum* (the smallest value of E). This is depicted in the following Figure 3-4:

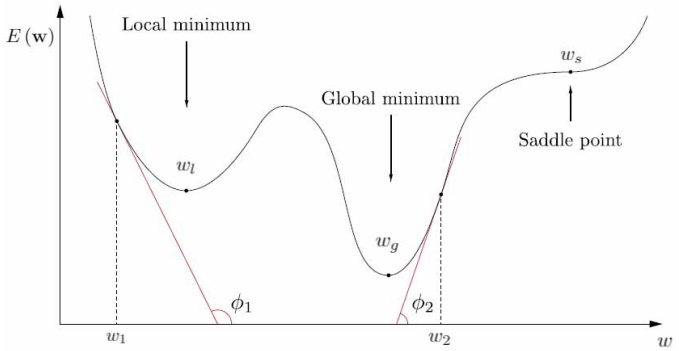


Figure 3-4: Local minima, saddle point and global minimum.(Adapted from [32])

The *Offline* approach has been found not robust to those points. In other words, training the ANN with the whole training data at once will cause the method will be more susceptible to find a *local minima* and stay there.
On the other hand, we can update $\mathbf{w}$ for each data point at time. It means, for each data point we make the feed-forward and back propagation process and update (3-10). This approach is called Stochastic or Online Gradient Descent (SGD). This approach has been found very robust for escaping local minima. Notwithstanding, using the SGD approach could increase the training time of an ANN sharply.
Thus, an intermediate approach is usually taken where the training set is split into smaller groups of data points, and the process of feedforward and backpropagation is applied to these subgroups. This approach is called *Mini-Batch* GD. This approach is a trade-off between faster convergence and keeping the algorithm fairly robust to escape a local minima.
In this report, the last approach is used.

3.7 Generalization and Overfitting

In section 3.2 it was stated that an ANN can represent a nonlinear function with a desired accuracy provided enough neurons using just one hidden layer. Nevertheless, experience has shown that a model can be more complex(in this case, it can have more neurons) only if the data set is larger. This is illustrated by the *Learning Curve* in the following Figure 3-5:

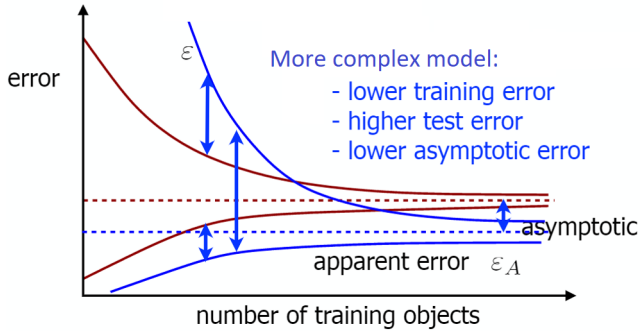


Figure 3-5: Empirical relation between error,model complexity and size of the data set.(Adapted from [32])

From the above Figure 3-5, we have that if the model is too complex(blue line) and the number of training objects (size of the dataset) is small, the error of the model for a test set(unseen data for the model) will be higher than the error of the training set(data which was used in order to train the model). In other words, the model will try to fit the training data almost perfectly(biased to the training set), which will diminish their power of generalization for unseen data(high variability). This phenomenon is called *Overfitting*, and it is illustrated by Figure 3-6:

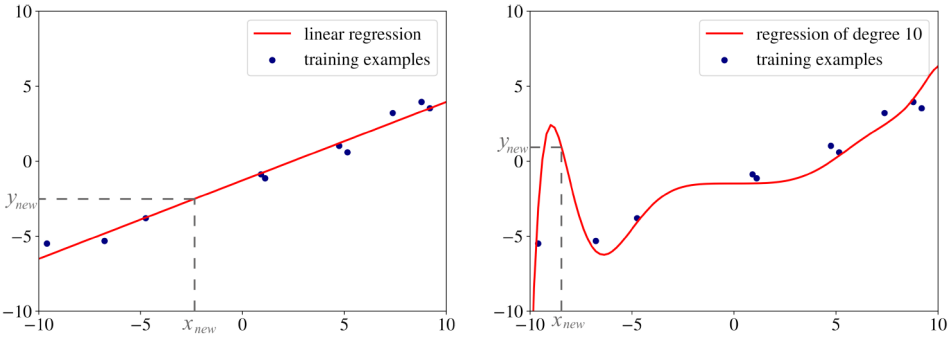


Figure 3-6: Examples of good fitting (at the left figure) and Overfitting due to a complex model (at the right figure).(Adapted from [4])

From Figure 3-6, we have at the left figure that the data points (blue dots) have a linear behaviour and a linear regression causes a higher sum-of-squares error (not perfect fitting), but the model captures the underlying structure of the data points. Conversely, in the figure at the right, the sum-of-squares error is decreased (almost perfect fitting), but the model is unable to capture the behaviour of the system (more variability). This occurs because, in

the second case, the model is too complex (polynomial of degree 10) for the number of data points (10 data points).
Thus, in order to prevent this undesired behaviour, the data set is usually split into three subsets: *training*, *validation* and *test set*. The first one is used to fit the model, the second is used while the model is being fitted in order to check if our model is suffering *overfitting* and the third one is used to test our model after the training procedure has been completed.
Additionally, there are some techniques in order to avoid overfitting. One of the most common approach is called *Early Stopping*, which is based on the idea of training our model until the error in the validation test (also called *validation or generalization error*) starts increasing. This is because the number of epochs that must be completed in order to achieve a desired error is a priori unknown (it is, in fact, a user-defined parameter), training the ANN over many epochs can lead to very small error which can be a sign of *overfitting* as it can be seen in the following Figure 3-7:

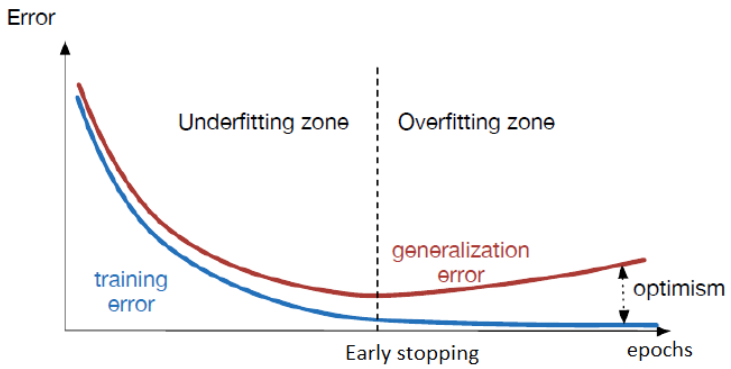


Figure 3-7: Schematic description of *Early Stopping*.(Adapted from [8])

Although there are other useful techniques such as dropout, batch normalization, adding artificial noise and other in order to avoid *overfitting*, the *Early Stopping* technique will be used in this report.

3.8 Advantages of using ANN

From the previous sections, we can observe the following advantages of using ANN for the current project, which are given as follows:

- Low memory requirement: This is because, after the training process of an ANN, only the architecture structure and the adaptive parameters need to be stored. Other machine learning methods must store part of the training data, making them unfeasible for the enormous amount of data points store in the lookup tables(order of millions points). Additionally, this will avoid storing computational expensive lookup tables.
- Smooth representation: This is due to the use of neurons and hidden layers, which, in principle, may allow approximating any continuous function from a finite-dimensional

space to another with any desired non zero error providing enough hidden neurons (see section 3.2).

- Great capability to handle large data sets: During the training process of an ANN, there is no need to compute and store any matrix and its inverse. The *Feedforward* and *Backpropagation* algorithms (section 3.5) are based on addition , subtraction and computation of inner products between vectors. This way of learning is the major advantage of an ANN with respect to other Machine Learning (ML) techniques. Additionally, this is why it is so popular its use in many problems.

Chapter 4

Look-Up tables

In this chapter, it will be explained the Flamelet Generated Manifold (FGM) approach used in order to generated the look-up tables. Additionally, the look-up tables that we aim to simplify in the current project will be introduced.

4.1 Flamelet Generated Manifolds approach(FGM)

The *FGM* approach proposed by Van Oijen et al.[21] relies on the assumptions that there is a fixed flame structure (flamelet) and the evolution of a scalar quantity in a turbulent flame can be locally represented by the evolution in a laminar flame. Thus, it shares the idea of the flamelet approach that a multidimensional flame can be modeled as an ensemble of one-dimensional flames[21]. In this way, a flamelet is defined by the set of equations given as follows [29]:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = -\rho K, \tag{4-1}$$

$$\frac{\partial \rho Y_i}{\partial t} + \frac{\partial \rho u Y_i}{\partial x} = \frac{\partial}{\partial x}\left(\rho D_{im} \frac{\partial Y_i}{\partial x}\right) + \dot{\omega}_i - \rho K Y_i, \tag{4-2}$$

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u h}{\partial x} = \frac{\partial}{\partial x}\left[\frac{\lambda}{C_p}\frac{\partial h}{\partial x} + \sum_{i=1}^{N_s}\left(\rho D_{im} - \frac{\lambda}{C_p}\right) h_i \frac{\partial Y_i}{\partial x}\right] - \rho K h, \tag{4-3}$$

$$\frac{\partial \rho K}{\partial t} + \frac{\partial \rho u K}{\partial x} = \frac{\partial}{\partial x}\left(\mu \frac{\partial K}{\partial x}\right) + \rho_{ox} a^2 - \rho K^2, \tag{4-4}$$

Where $K = \frac{\partial v}{\partial y}$ represents the stretch term. Likewise Y_i, h, ρ, D_{im}, $\dot{w}_i$, λ, C_p and μ denote the mass fraction of species, the specific enthalpy of the mixture, the density, the mixture mass diffusion coefficient, the reaction rate, the thermal conductivity, the heat capacity and the viscosity, respectively. Finally, a and ρ_{ox} are the applied strain rate and the density at the oxidizer side.

Note that these flamelet equations include the effect of differential diffusion which will be discussed in section 4.2. When the effect of differential diffusion is not important, it can be

assumed that all species have the same diffusivity (equal Lewis number), then we have that $D_{im} = D$ for all species and the energy equation 4-3 becomes in:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u h}{\partial x} = \frac{\partial}{\partial x}\left(\frac{\lambda}{C_p}\frac{\partial h}{\partial x}\right) - \rho K h, \tag{4-5}$$

Commercial and open source one-dimensional flame codes which use the ideal gas equation of state have been developed in the last years, such as CHEM1D[30] developed by TU Eindhoven which is used in this research.
Thus, in the FGM approach a low dimensional manifold is constructed from one-dimensional steady and unsteady flamelets[21] where they are expressed as functions of the mixture fraction Z and the scaled progress variable C. The mixture fraction is defined as the fraction of mass present locally that originally comes from the fuel stream (it measures the local fuel/oxidizer ratio [25]). Thus, the mixture fraction is a key variable to depict mixing in non-premixed (diffusion) flames. Likewise, the scaled progress variable depicts the combustion process because it goes from 0 in the urburnt gases to 1 in the fully burnt gases. Thereby, the choice of the progress variable is crucial in this approach because it must provide a unique mapping for all species $Y_i(C)$ [34]. Hence, it must be a monotonously increasing or decreasing function[34]. In this way, the unscaled progress variable Y_C is defined as a linear combination of a subset of species mass fraction and the scaled progress variable C are given by the following expressions:

$$Y_C = \sum_{1}^{N_s} \alpha_i Y_i \tag{4-6}$$

$$C = \frac{Y_C - Y_{C,min}}{Y_{C,max} - Y_{C,min}} \tag{4-7}$$

where the subscripts min and max represent the minimum and maximum values of the Y_C, respectively. In order to ensure that C is monotonously increasing, Ramaekers[26] proposed to define α_i as the inverse of the molar mass of the species i, which is used in the creation of the lookup tables considered in this work.
In the turbulent flames, the influence of statistical fluctuations is included considering the variance of the mixture fraction $\widetilde{Z''^2}$ and the variance of the scaled progress variable $\widetilde{C''^2}$. These are included as independent variable in order to characterise a joint Probability Density Function (PDF) where C and Z are assumed to be statistically independent. A standard PDF choice for mixture fraction and the scaled progress variable is a β-distribution. Other independent variables can be also added in the manifolds in order to take into account other effects such as heat loss (Non-adiabatic flame), dilution and other, as will be the explained in section 4.3.
In sum, a look-up table is built where dependent variables (reactions rates, mass fractions, temperature and others) are stored as a function of a set of reduced independent variables (mean mixture fraction, mean scaled progress variable their respective variances and others). It must be noted that for each independent variable, a transport equation is solved during Computational Fluid Dynamics (CFD) simulations and the information stored in the look-up tables is retrieved during the simulations.
Finally, this approach has been demonstrated to depict the behaviour of low temperature areas in a better way than the Intrinsic Low Dimensional Manifold (ILDM) approach proposed in

[16]-[17], due to the fact that it takes into account transport processes which are not included by the aforementioned approach[34]. Although for many steady flames this drawback does not cause issues because chemical processes are principally controlled by the highest temperatures [25], for other problems such as ignition, transient and diffusion phenomena it may cause difficulties[25].

4.2 4D hydrogen look-up table

In the current context of climate change, searching for fuels that have low carbon emissions has become important in order to achieve a sustainable growth in the future. That is the reason why hydrogen seems a promising fuel. However, in order to be able to use hydrogen in a wide range of applications, models must be more accurate in order to obtain better and reliable predictions. In turbulent combustion, a unit Lewis number (which compares the speeds between thermal and species diffusivity[25]) is commonly adopted due to the fact that species have similar diffusivities. This relies on the assumption that molecular diffusion is negligible with respect to turbulent mixing. However, that is no longer valid in the case of hydrogen where its Lewis number is around 0.2. Thus, it has been been reported in [6] that the effects of differential diffusion are important close to the nozzle while Large Eddy Simulations (LES) in H3 flame were conducted. Similarly, its effects on maximum flame temperature , flame position and stability were reported in [18]. Thereby, differential diffusion must be taking into account in hydrogen flames.
This was done by Myra Nelissen [20] during her master thesis project at TU Delft where she included differential diffusion in the FGM approach. In her project, the unscaled progress variable was defined as the mass fraction of water ($Y_C = Y_{H_2O}$) and the mixture fraction Z was defined using the Bilger's mixture fraction based on H and O given as follows:

$$Z = \frac{z_{un} - z_{un,2}}{z_{un,1} - z_{un,2}} \tag{4-8}$$

$$z_{un} = 0.5 M_H^{-1} Y_H - M_O^{-1} Y_O \tag{4-9}$$

Where M_H, M_O and Y_H, Y_O are the molar mass and mass fraction of hydrogen and oxygen, respectively. Likewise, the subscripts un denotes unscaled mixture fraction and 1 and 2 represents the fuel and oxidizer side, respectively.
Finally, the chemical mechanism for H_2/O_2 proposed by Li et al.[14] consisting in 19 reversible reactions, 9 species and nitrogen as dilutant was considered. Thus, 4D Lookup tables for turbulent flames were created, where the 4D independent variables are: mean scaled progress variable ($\widetilde{C}$), mean mixture fraction ($\widetilde{Z}$) and their respective variances ($\widetilde{C''^2}$,$\widetilde{Z''^2}$). Note that $\widetilde{\cdot}$ denotes the Favre Averaging.

4.3 6D Diluted-Air Flamelet generated Manifolds table(DA-FGM)

Although we are experiencing a transition into hydrogen technology, fossil fuels will still be mainly used in the coming decades. Hence, better models are needed in order to optimize their use and reduce their emissions. In this way, the FGM approach can be extended to Moderate or Intense Low Oxygen Dilution (MILD) combustion as long as the flamelets consider the

effects of strong dilution by products. This was done by Xu Huang[9] during his PhD project at TU Delft and the accepted article[10]. In them was developed an extension of the FGM approach called Diluted Air FGM (DA-FGM) which includes recirculation of burnt gases to the reaction zone (dilution) and was validated by experimental data obtained using Dutch natural gas in a lab-scale furnace. This six-dimensional lookup table had not been studied before and calculations on spectral radiative heat transfer were performed in specific profiles along lines, finding spectral effects to be important. This suggest that in the case of spectral treatment more variables have to be stored in the tables, making them unfeasible in CFD simulations due to lack of memory, which is directly linked with the goal of the current project to reduce substantially the memory requirement. Although it was only possible to compute RANS numerical simulations using it due to the large memory requirement of this table, high accuracy was achieved employing the lookup table generated by DA-FGM.
Now, we will proceed to explain the main assumptions and definitions that underlies in the DA-FGM approach.
First, in MILD combustion the reacting flow structure is assumed to be a mixture between fuel, air and burnt gases. Considering a global equivalence ratio of $\phi = 1$ and a long residence time in the furnace, the recirculated burnt gases can be assumed as products at Z_{st} and being in chemical equilibrium. Therefore, the diluent is defined as products of combustion of fuel and air at stoichiometric conditions (without excess).
Subsequently, in order to introduce the effect of dilution in the flamelets, two variables are defined:

- Second mixture fraction ξ which is the mixture fraction related to the fuel-diluted air mixing. In the DA-FGM model, the mixture is based on fuel and diluted air streams instead of the regular mixture fraction Z defined by fuel and air streams. Hence, as the diluent is at stochiometric mixture fraction, ξ is defined as follows:

$$\xi = Z - \alpha Z_{st} = (1 - \alpha) Z_0 \tag{4-10}$$

  where Z_0 is the mixture fraction of fuel-air mixture. Additionally, α is called dilution level which represents the fraction of diluent in the mixture. This is defined by:

$$\alpha = \frac{Y_d}{Y_d^{Dil}} \tag{4-11}$$

  where Y_d is the mass fraction of recirculated products in fully burnt condition. Thus, Y_d is defined by

$$Y_d = Y_{CO_2} + Y_{H_2O} \tag{4-12}$$

  and Y_d^{Dil} denotes the mass fraction of recirculated products at stoichiometric condition.

- Air dilution level γ which is the mass fraction of diluent in the diluent/air mixture. This is computed by the following expression:

$$\gamma = \frac{\alpha}{1 - \xi} \tag{4-13}$$

It must be noted that $\xi = 1$ and $\xi = 0$ imply pure fuel state and diluted air state, respectively.

For Non-adiabatic flamelets, an additional variable is introduced in order to reflect the heat loss effects. So, the enthalpy loss factor η is introduced. Due to radiation and heat transfer to cold walls losses, the diluent has an enthalpy loss with respect to the adiabatic case. Thus, η is defined by:

$$\eta = \frac{h - h_{ad}}{\alpha \left(h^d_{\eta=1} - h^d_{\eta=0} \right)} \tag{4-14}$$

where h_{ad} is the adiabatic enthalpy at local mixture fraction given by:

$$h_{ad} = Z h_f + (1 - Z) h_{ox} \tag{4-15}$$

where the subscripts f and ox denotes fuel and oxidizer side. Additionally, $h^d_{\eta=1}$ and $h^d_{\eta=0}$ are the enthalpy of diluent in the case of maximum and minimum enthalpy loss, respectively. Finally, the unscaled progress variable Y_C used in this model is given by :

$$Y_C = Y_{CO_2} + Y_{CO} + Y_{H_2O} + Y_{H_2} \tag{4-16}$$

and the scaled progress variable C is computed as follows:

$$C = \frac{Y_C - Y^u_C \left(\xi, \eta, \gamma \right)}{Y^b_C \left(\xi, \eta, \gamma \right) - Y^u_C \left(\xi, \eta, \gamma \right)} \tag{4-17}$$

where u and b denote unburnt and burnt state, respectively. It must be noted that the minimum and maximum values of the unscaled progress variable (Y^u_C and Y^b_C, respectively) depend on the local conditions in the flamelets. Therefore, they depend on ξ, η and γ. Thereby, they are also precomputed and stored in a 3D lookup table.
In sum, in the laminar case we have four independent variables given by equations (4-10),(4-13), (4-14) and (4-17).
On the other hand, for turbulent flames the variance of the second mixture fraction $\widetilde{\xi''^2}$ and the variance of the scaled progress variable $\widetilde{C''^2}$ are introduced as additional independent variables assuming a β-PDF for mixture fraction and scaled progress variable. Then, they are normalized as follows:

$$S_\xi = \frac{\widetilde{\xi''^2}}{\widetilde{\xi} \left(1 - \widetilde{\xi} \right)} \tag{4-18}$$

$$S_C = \frac{\widetilde{C''^2}}{\widetilde{C} \left(1 - \widetilde{C} \right)} \tag{4-19}$$

So, the thermochemical quantities such as temperature, source term of the progress variable and others are stored as function of the following six independent variables: $\widetilde{\xi}, \widetilde{C}, S_C, S_\xi, \widetilde{\eta}$ and $\widetilde{\gamma}$.
According to the standard procedure, to use this 6D lookup table transport equations are solved for the variables $\widetilde{Z}, \widetilde{Y_C}, \widetilde{Y_d}, \widetilde{h}, \widetilde{Z''^2}$ and $\widetilde{Y''^2_C}$. Subsequently, the six independent variables

are computed using the following expressions:

$$\widetilde{\xi''^2} = \widetilde{Z''^2} \tag{4-20}$$

$$\widetilde{C''^2} = \frac{\widetilde{Y_C''^2} - \left(\widetilde{Y_C}\right)^2 - \widetilde{(Y_C^u)^2} - 2\widetilde{C}\left[\widetilde{Y_C^u Y_C^b} - \widetilde{(Y_C^u)^2}\right]}{\left(\widetilde{Y_C^b} - \widetilde{Y_C^u}\right)^2} - \widetilde{C}^2 \tag{4-21}$$

$$\widetilde{\xi} = \widetilde{Z} - \widetilde{\alpha} Z_{st} \tag{4-22}$$

$$\widetilde{\gamma} = \frac{\widetilde{\alpha}}{1 - \widetilde{\xi}} \tag{4-23}$$

$$\widetilde{\eta} = \frac{\widetilde{h} - \widetilde{h}_{ad}}{\widetilde{\alpha}\left(h^d_{\eta=1} - h^d_{\eta=0}\right)} \tag{4-24}$$

$$\widetilde{C} = \frac{\widetilde{Y_C} - \widetilde{Y_C^u}}{\widetilde{Y_C^b} - \widetilde{Y_C^u}} \tag{4-25}$$

Where, the quantities $\widetilde{Y_C^b}$, $\widetilde{Y_C^u}$, $\left(\widetilde{Y_C}\right)^2$, $\widetilde{(Y_C^u)^2}$ and $\widetilde{Y_C^u Y_C^b}$ are precomputed in a 4D-lookup table as a function of the variables $\widetilde{\xi}$, S_ξ, $\widetilde{\eta}$ and $\widetilde{\gamma}$.
Likewise, it must be pointed out that no fluctuations in the dilution level α are considered, so it is computed as follows:

$$\widetilde{\alpha} = \frac{\widetilde{Y_d}}{Y_d^{Dil}} \tag{4-26}$$

Additionally, the DA-FGM approach includes radiation using a Weighted-Sum-of-Grey-Gases (WSGG) model in order to study Turbulent Radiation Interaction (TRI). The main assumptions behind this model choice are that absorption and emissions must be taken into account in gas fired furnaces, meanwhile scattering can be neglected. In this way, the grey gas absorption coefficient κ is obtained by using the WSGG model considering four grey gases and one clear gas. This coefficient is used in order to solve the mean Radiative Transfer Equation (RTE) given as follows:

$$\frac{d\overline{I^m}}{ds} = -\overline{k} \cdot \overline{I^m} + \overline{\kappa I_b} \tag{4-27}$$

Where I and I_b denotes the radiative intensity and blackbody radiation intensity, respectively. Aditionally, m refers to spectral band. The terms $\overline{k}$ and $\overline{\kappa I_b}$ depends on local conditions of the flames. Therefore, they are stored in the 6D lookup table. Solving equation (4-27) allows coupling radiation with combustion in the transport equation solved for enthalpy $\widetilde{h}$. For more details, see [9] and [10].
Finally, the GRI-Mech 3.0 reaction mechanism, which models natural gas combustion, is used in the generation of the 6D lookup table for Dutch natural gas. This consists of 325 reactions and 53 species.
As a final remark, from the previous equations (in particular, equation (4-23)), it can be observed that the six independent variables in this lookup table are not completely independent between each other.

Chapter 5

Data and Numerical setup

This chapter will explain the data that will be used, the numerical setup and the software that have been used in order to develop the Artificial Neural Network (ANN).

5.1 Data

The data used in the research consists of the 4D and 6D lookup tables. The tables are generated as follows: First, steady and unsteady flamelets are generated using CHEM1D[30]. Subsequently, a FGM table is created using a Matlab code. Finally, PDF-integration is performed, and a FGM-PDF table is created using a Matlab code, where separate files are creating for storing different variables (outputs). The Matlab codes were developed by Mengmeng Ren, Likun Ma and Xu Huang.
The information contained in each file is tabulated in a regular grid as a function of the independent variables (four and six independent variables for the 4D and 6D lookup table, respectively). The variables and number of nodes in each direction for the 4D and 6D lookup tables are given in the following tables 5-1 and 5-2:

Table 5-1: Independent variables and number of nodes in each direction in the 4D lookup table.

Variable	Number of Nodes
Mean mixture fraction	101
Variance of the mixture fraction	11
Mean scaled progress variable	101
Variance of the scaled progress variable	11

Table 5-2: Independent variables and number of nodes in each direction in the 6D lookup table.

Variable	Number of Nodes
Mean mixture fraction	51
Variance of the mixture fraction	11
Mean scaled progress variable	51
Variance of the scaled progress variable	11
Scaled enthalpy loss factor	13
Scaled dilution variable	11

Then, a single Comma Separated Value (CSV) file for each lookup table is created, containing the inputs in a regular grid. For the 4D lookup table, the input file contains $11 \cdot 101 \cdot 11 \cdot 101 = 1234321$ nodes. Similarly, for the 6D lookup table, the input file contains $51 \cdot 11 \cdot 51 \cdot 11 \cdot 13 \cdot 11 = 45005103$ nodes.
Additionally, each output file, which is in text format, is brought in CSV format. The CSV format is required because Pandas library for handling large data sets will be used, and the files must be in that specific format. The CSV files are created using a Python routine.

5.2 Pre-processing data

First of all, the inputs (x_i) in each table are in the interval $[0, 1]$. Therefore, they do not require to be rescaled. Outputs (y_i) have to be rescaled using the following expressions:

$$\widehat{y}_i = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \tag{5-1}$$

Where y_{min} and y_{max} correspond to the minimum and maximum value in the look-up table for each output. This is done in order that inputs and targets have the same scale, in this way the adaptive parameters ($\mathbf{w}$) can capture successfully the importance of each feature for the target (output) y_i.

5.3 ANN architectures

As it was explained in section 3.2, there is no rule of how to choose the number of hidden layers and the number of neurons per hidden layer. Hence, we are going to make this decision using the following procedure:

1. In [32], it is strongly recommended to review what others have done in similar projects (learning through experience). Similarly, although the Universal approximation theorem (UAT) states that with one single hidden should be enough for regression problems, it is well-known from mathematics theorems (theorem of change of variables, chain rule and others) that using successive mappings may make a problem much easier to solve. Therefore, it is adviced to use more than one single layer. Additionally, using more than one hidden layer may reduce the memory requirement of the ANNs.

2. Similarly, following the *Kolmogorov* theorem, it would be enough to use an ANN with two hidden layers, the first layer with $2 \cdot (2 \cdot 4 + 1) = 18$ neurons and the second hidden layer with $2 \cdot 4 + 1 = 9$ neurons, in the case of the 4D Look-up table $d = 4$ (Similar for the 6D Diluted Air FGM (DA-FGM) look-up table for which $d = 6$). Nevertheless, from previous works [11],[35] and [27], it can be concluded that two hidden layers are not enough for representing the look-up tables with high accuracy.

3. As the UAT and Kolmogorov theorem are valid for a one-dimensional output function (see section 3.2), they implicitly suggest that using one ANN per output would provide better results. Therefore, that approach is considered in this project.

4. Thus, as [11] and [27] considered in their works 4 hidden layers and other even more hidden layers [35], we start using 4 hidden layers.

5. Additionally, in the projects aforementioned, they used between 8 and 10 neurons per layers. As our data set is larger than the ones considered in those projects and, as the input dimensionality is higher than the works of [11] and [35], we increase the number of neurons per hidden layer. For most of the species in the 4D table, we found out that the following architecture provides good results: 32-16-8-4 (865 parameters). Where each number represents the number of neurons in each layer (32 neurons in the first hidden layer and so on). This architecture was chosen in order to have a common architecture for most of the species and thermochemical quantities. Likewise, it provides accurate results and stability as it will be shown in section 6.1.

6. For the source term of the progress variable, the architecture used was 16-16-16-16 (913 parameters).

7. In the 6D DA-FGM look-up tables, the data size is higher than the 4D case. Similarly, the input dimensionality is higher (6 independents variables). Hence, the number of neurons per layer must be increased in order to obtain high accurate and stable results[7].

8. Thus, in the 6D case, we start training an ANN for the source term of the Progress variable in the 6D lookup table because it is well-known that it is a highly non-linear function. With the knowledge gained from the 4D lookup table. First an ANN architecture with 20 neurons per layer (1421 parameters) is used, then this number is increased by 4, and an ANN with 24 neurons per layer (1993 parameters) is trained. As the data set is large, the number of neurons is increased to 30 neurons per layer (3031 parameters), which provides the most accurate and stable results.

9. Considering that the rest of the thermochemical quantities should be, in principle, easier to represent using ANNs than the source term of the progress variable. We reduce the number of neurons per hidden layer to 26, which provides accurate and stable results for these outputs.

Additionally, as we are using an ANN for regression, we first consider in the hidden layers Rectified Linear (RELU) as the activation function and Sigmoid as the activation function at the output layer (see section 3.1, expressions (3-1),(3-3)). Nonetheless, as the preliminary results using this activation function are unstable, we consider Sigmoid(equation (3-3)) as activation function in the hidden and output layers. This choice provides stable results and,

at the same time, force the ANN to respect the physics (predictions will not be bigger than the biggest values in the tables and they will not be negative).

5.4 Training Procedure

The training procedure is explained as follows: For each architecture, we will train the ANN using the mini-batch GD with size of 16 and 128 data points for the 4D and 6D look-up tables, respectively.
Likewise, for each architecture, 200 *epochs* will be completed in order to obtain accurate results.
Furthermore, as machine learning methods are susceptible to *overfitting*(see section 3.7), we split the data set into three subsets: *training set*, *validation test* and *test set*. There is no rule about in which way the data set must be split. In this report, we split randomly the data set as follows: 70% as *training set*, 15% as *validation set* and 15% as *test set* with entries put in these subsets by random selection.
Equally important is the choice of the learning rate (μ) because it is a *hyper-parameter* of the model that controls how the models are learning a specific task. Hence, it must be chosen carefully. Notwithstanding, it is very difficult to know apriori which value of μ is the most accurate for a given data set. Therefore, an adaptive learning rate is used instead of a fixed learning rate. This adaptive learning rate works as follows: First, we consider an initial learning of $\mu = 0.005$, and it is checked the model improvement observing the value of the loss function (the Mean Squared Error (MSE)) in the validation set. If the loss function in the validation set does not improve after 5 epochs, the learning rate will be reduced by a factor of $f = 0.5$. This approach is also known as *ReduceLROnPlateau*.
Subsequently, each time that there is an improvement of the loss function in the validation set, the model is saved (best model). Finally, if after 30 epochs the model does not improve in the validation set (a sign of overfitting), the iteration will be stopped (Early Stopping, see section 3.7).

5.5 Software

The ANNs have been developed using Python 3.8.5. The environment used is *Jupyter lab*. Additionally, scikit-learn[23], numpy, pandas, and others libraries available in Python have been used. Finally, *Keras* which is a deep learning application programming interface (API) has been used to build the *ANN* . This is running on the open-source platform for machine learning *TensorFlow*[31].

Chapter 6

Results and Discussions

In this chapter, the results of this project will be presented. First, results regarding the choice of the activation function used in the ANN (see section 3.1) will be discussed. Subsequently, preliminary results of using ANNs for the 4D hydrogen lookup table will be presented and the unexpected issues encountered in this table will be pointed out. Then, a solution to these issues is proposed, and the final ANN models for the 4D lookup table will be shown. Finally, the results of ANNs applied for the 6D DA-FGM table will be analysed.

6.1 4D Hydrogen Table

We follow the procedure explained in the previous chapter 5. As the 4D lookup table is smaller in size than the 6D lookup table, we begin with this case and make preliminary simulations in order to apply the knowledge gained for the easier case into the 6D lookup table, which in principle is more difficult than the previous one.

6.1.1 Preliminary Results

First, we start studying the most suitable choice concerning ANN architecture for the current project.

6.1.1.1 Activation function choice

From the literature review (2.1), we observe that Pitsch et al.[11] and Zhang et al.[35] used as activation function RELU (equation (3-1)) in the hidden layers and Sigmoid (equation (3-3)) in the output layer. Thus, in order to study the more suitable choice of activation functions for this project, we run a preliminary case where an ANN for the following output is trained: Source term of the progress variable (SourcePV). This output is chosen as a preliminary case because it is highly non-linear and it is, in principle, the most difficult output to be

represented by ANNs. Then, the ANN architecture for this preliminary case consists of 4 hidden layers and 10 neurons per layer. Finally, in order to visualise the results, we plot three different cases:

- Case 1: Case without fluctuation - the output varies with respect to the mean scaled progress variable $\widetilde{C}$, for different values of mean mixture fraction $\widetilde{Z}$, considering $\widetilde{C''^2} = \widetilde{Z''^2} = 0$.
- Case 2: Stoichiometric mixtures with fluctuations in mixture fraction - the output varies with respect to the variance of mixture fraction $\widetilde{Z''^2}$ at stoichiometric mixture fraction $\widetilde{Z} = \widetilde{Z_{st}}$, for different and high values of the mean scaled progress variable $\widetilde{C}$ and considering $\widetilde{C''^2} = 0$.
- Case 3: Stoichiometric mixtures with fluctuation in the scaled progress variable - the output varies with respect to the variance of the scaled progress variable $\widetilde{C''^2}$ at stoichiometric mixture fraction $\widetilde{Z} = \widetilde{Z_{st}}$, for different and high values of the mean scaled progress variable $\widetilde{C}$ and considering $\widetilde{Z''^2} = 0$.

Hence, we train the ANN for the SourcePV using the activation functions used in the papers mentioned above and considering Sigmoid in the output and hidden layers. In this way, we obtain the following results:

• RELU in hidden layers and Sigmoid in the output layer:

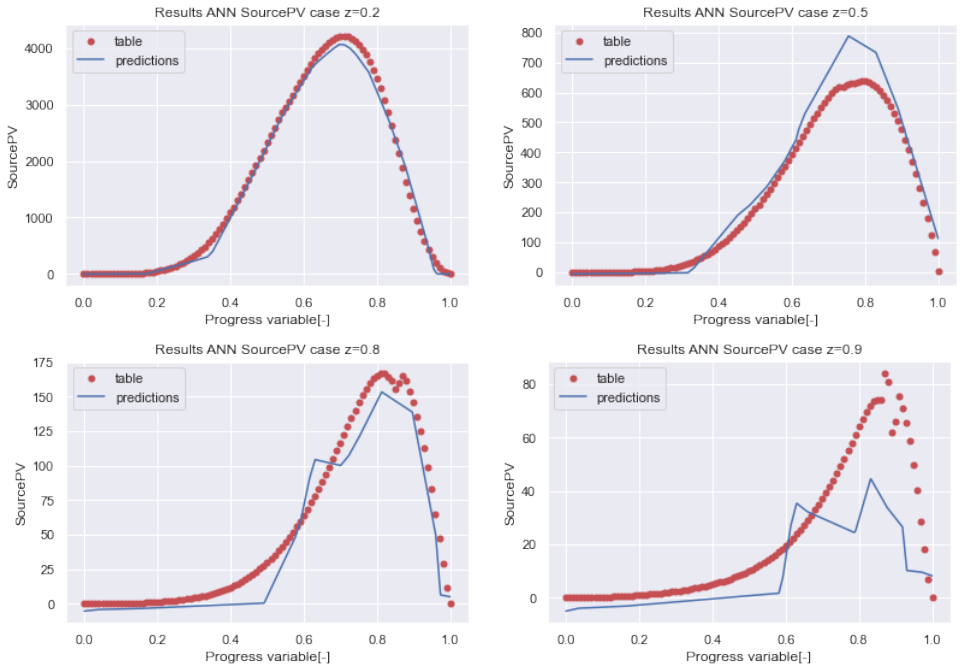


Figure 6-1: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction, using RELU and Sigmoid as activation functions in hidden and output layers, respectively.

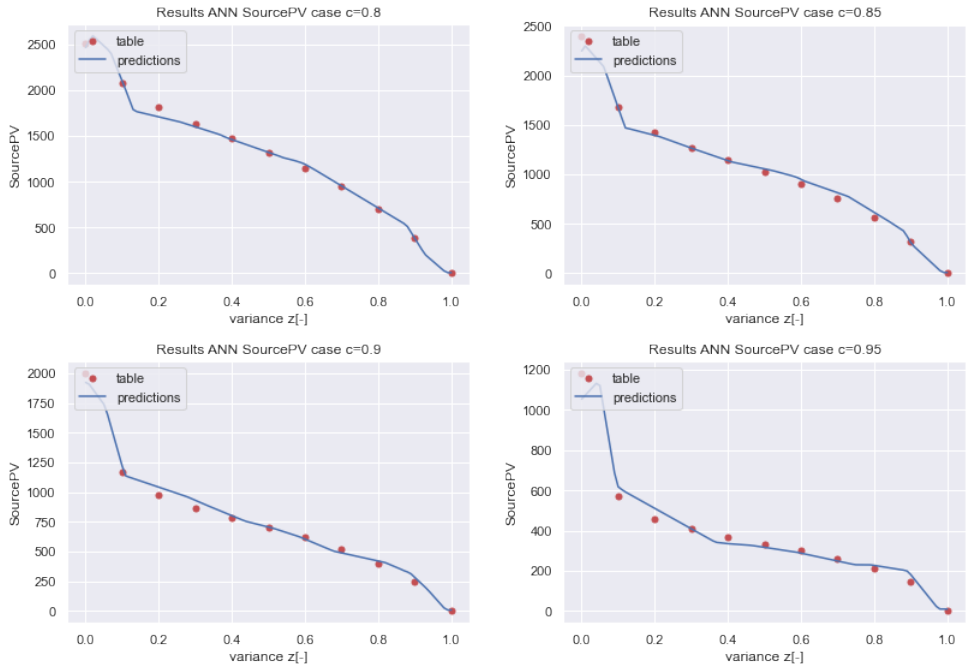


Figure 6-2: Case 2: Source term of the progress variable with respect to variance of mixture fraction for different values of mean scaled progress variable, using RELU and Sigmoid as activation functions in hidden and output layers, respectively.

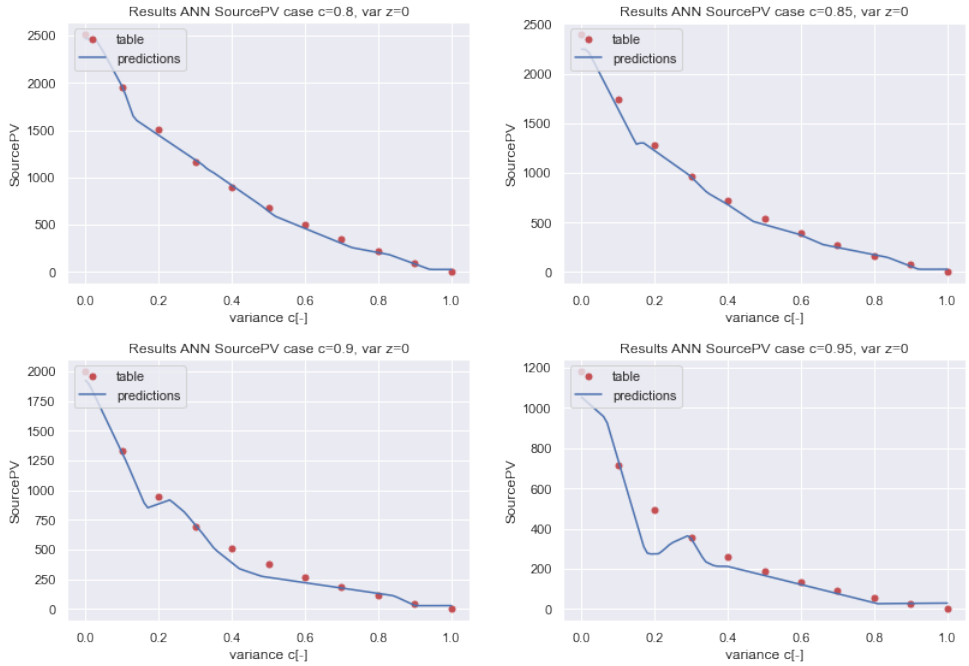


Figure 6-3: Case 3: Source term of the progress variable with respect to variance of the scaled progress variable for different values of mean scaled progress variable. Using RELU and Sigmoid as activation functions in hidden and output layers,respectively.

• Sigmoid as activation function in hidden and output layers:

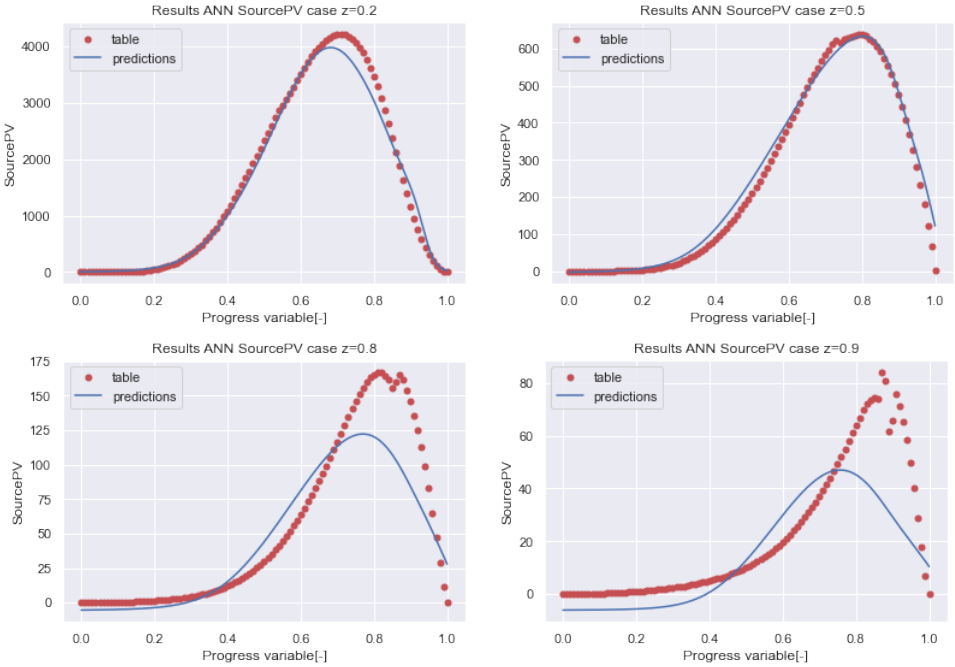


Figure 6-4: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction, using Sigmoid as activation function in hidden and output layers.

Table 6-1: Comparison of memory requirement and prediction(retrieval) times between two cases: RELU and Sigmoid as activation function in hidden and output layers, respectively, and Sigmoid as activation function in hidden and output layers. MSE_{train} and R^2_{train} denote the mean squared error and coefficient of determination for the training set, respectively.

Case	Memory (kB)	Prediction time[s]	MSE_{train}	R^2_{train}
RELU-Sigmoid	27	0.09794	0.000045	0.9908
Sigmoid	55	0.078287	0.000037	0.9924

From table 6-1, we can observe that both choices provide similar statistical indicators: Mean squared error MSE_{train} of order 10^{-5} and coefficient of determination R^2_{train} around 0.99. Similarly, the prediction/retrieval times using both choices are roughly equal. Hence, based purely on statistics, both choices are acceptable. Nevertheless, from figure 6-1, we can observe that using RELU-Sigmoid as it is done in [11] and [35] provides a model that is becoming more unstable for higher values of the mean mixture fraction. Moreover, we can see from figures 6-2 and 6-3 that the ANN model develops wiggles for higher values of the progress variable. Conversely, the previous instabilities are not observed when using Sigmoid as activation function, as shown in figures 6-4, 6-5 and 6-6. It also preserves the underlying physical behaviour of the source term of the progress variable as shown in figure 6-4. Therefore, for this project, we can conclude that the best choice in terms of stability and physical behaviour is to use Sigmoid as activation function in each layer.

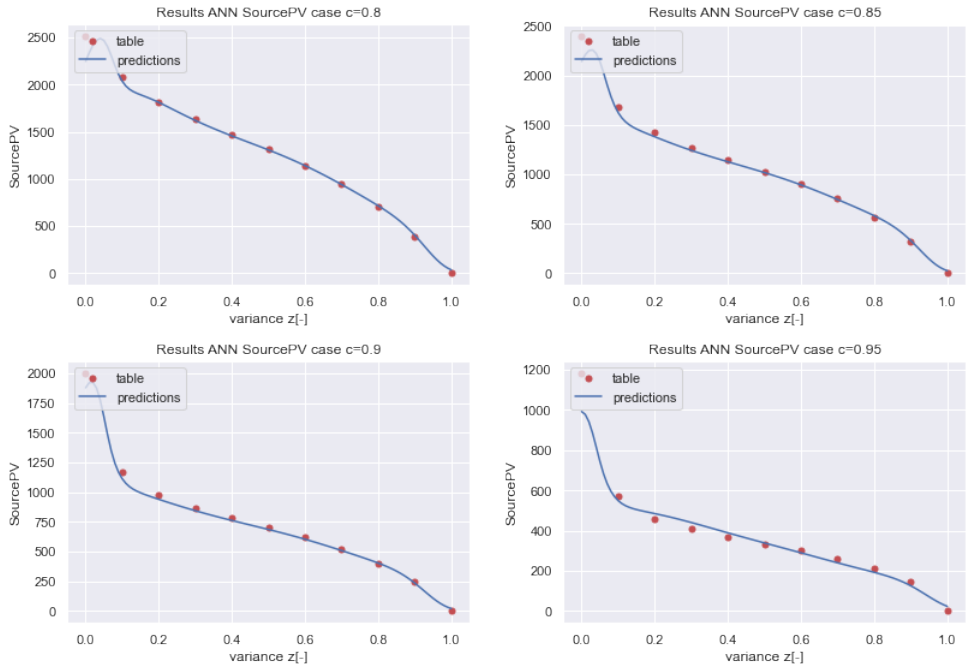


Figure 6-5: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable, using Sigmoid as activation function in hidden and output layers.

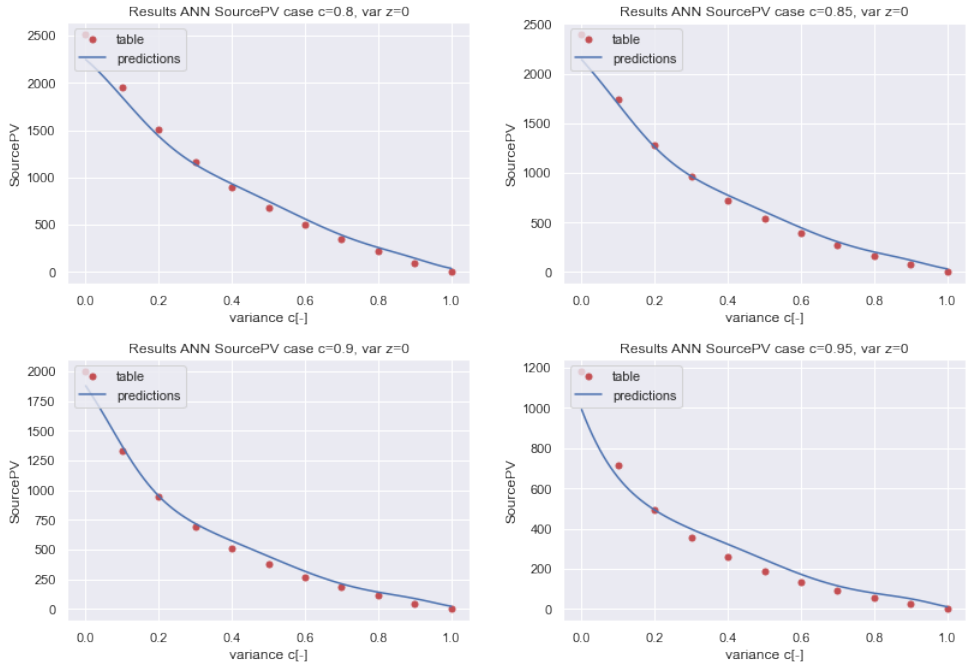


Figure 6-6: Case 3: Source term of the progress variable with respect to variance of the scaled progress variable for different values of mean scaled progress variable, using Sigmoid as activation function in hidden and output layers.

Although from table 6-1 we can observe that the memory requirement of using Sigmoid is more than twice the memory requirement of using RELU-Sigmoid, this memory requirement is still around 2.5 order of magnitude less than the lookup table(26.649 MB). Finally, in [27] hyperbolic tangent (equation (3-4)) is used due to its high non-linearity. However, using hyperbolic tangent would require rescaling the independent variables into the interval [-1,1] which is not the case with Sigmoid. This is because the Sigmoid is in the same range as the inputs (independent variables) $[0, 1]$. This makes it a better choice than the hyperbolic tangent because it would alleviate the task of coupling the ANNs models in a CFD solver, since only outputs would need to be rescaled not inputs.

6.1.1.2 ANN models for different outputs

From the analysis made in the previous section 6.1.1.1, we proceed using Sigmoid as activation function in hidden and output layer. Then, in order to train the ANN,we follow the procedure explained in section 5.4, and we find out that the best ANN architectures are: 16-16-16-16 for the Source term of the progress variable and 32-16-8-4 for the rest of the thermochemical quantities. In order to avoid a large amount of figures, we will show the results for the following outputs: Source term of the progress variable, temperature, heat capacity and mean molar mass of the mixture. Thus, training the ANN model for the outputs mentioned above, we obtain the following results where we have considered the cases 1,2 and 3 mentioned in section 6.1.1.1:

- Source Progress Variable[kg/(m^{3}s)]:

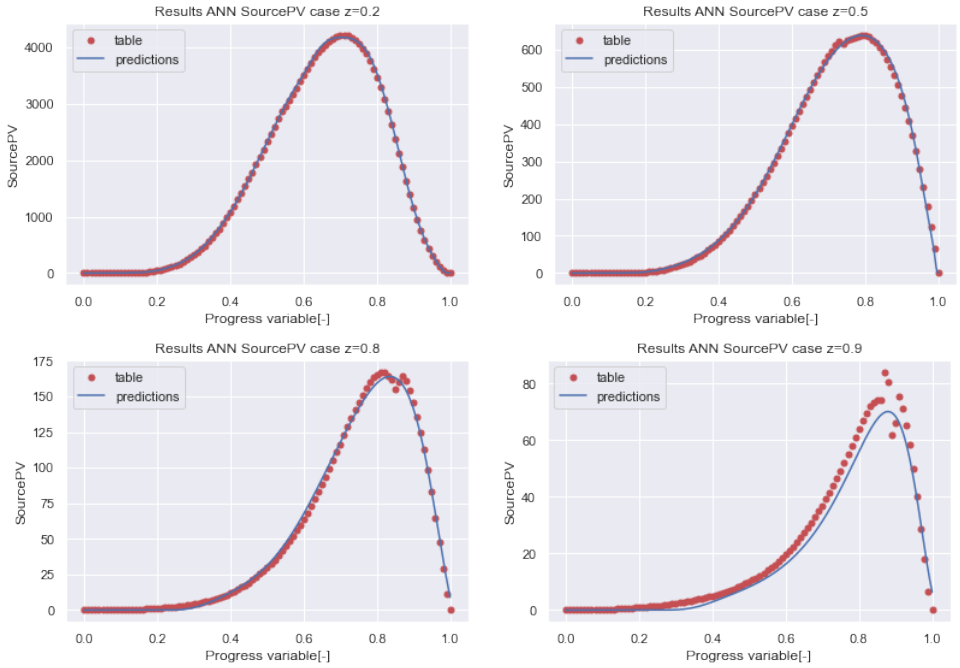


Figure 6-7: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction. Preliminary results.

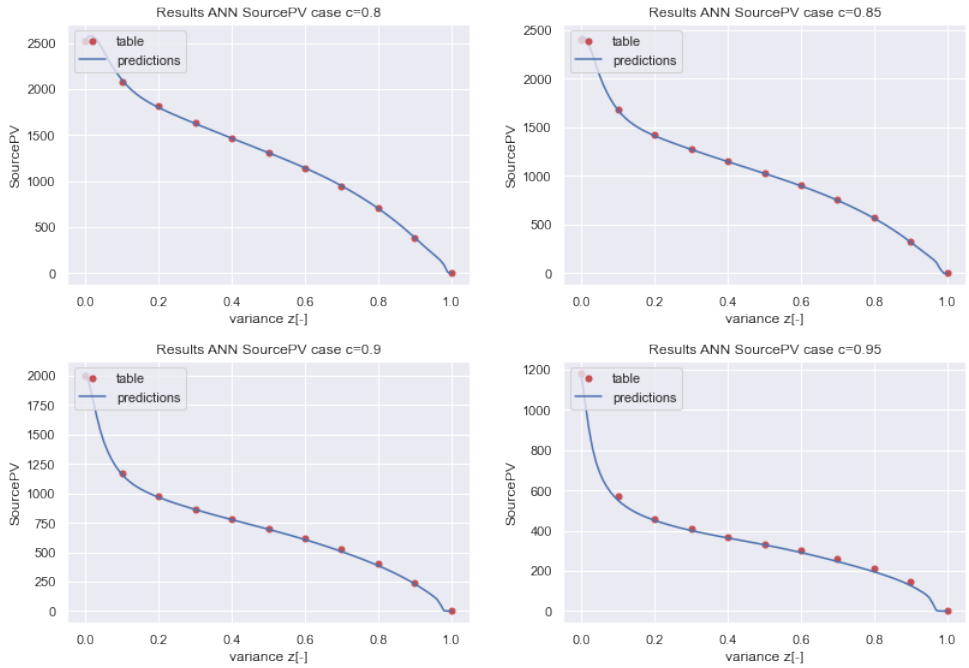


Figure 6-8: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable. Preliminary results.

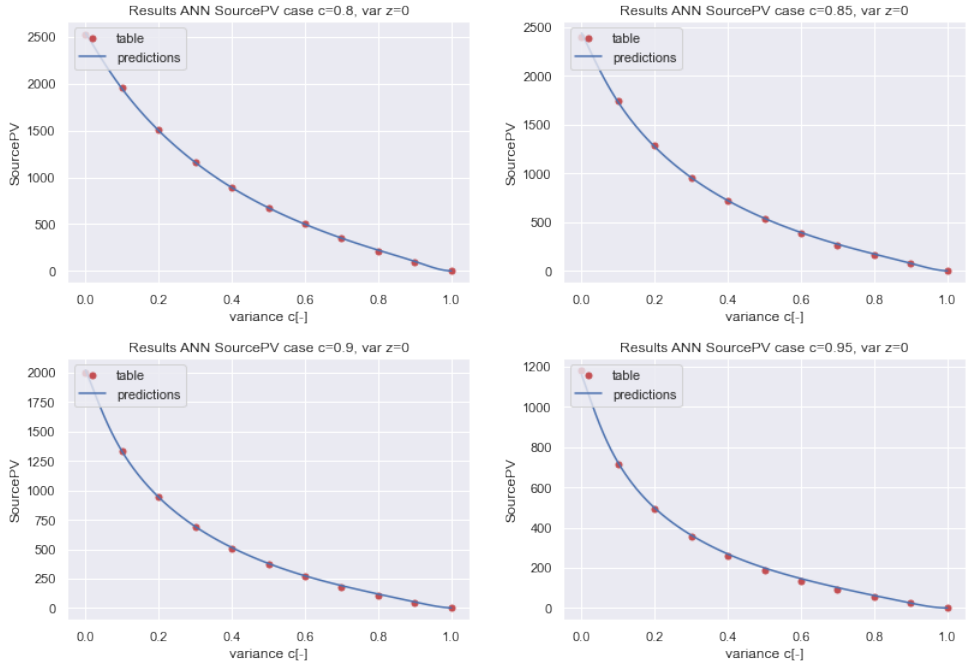


Figure 6-9: Case 3: Source term of the progress variable with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. Preliminary results.

• Temperature[K]:

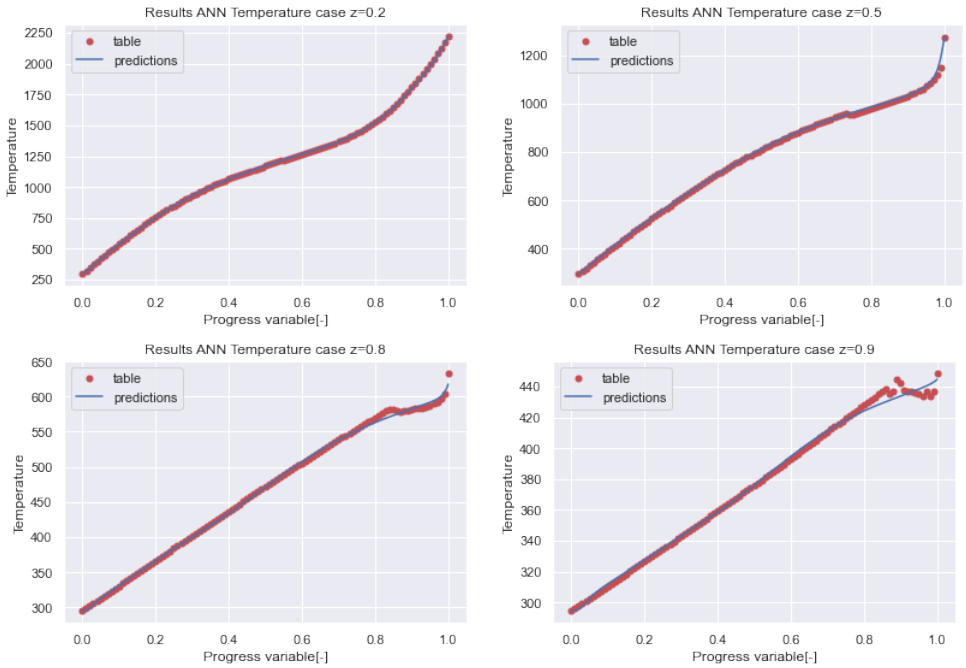


Figure 6-10: Case 1: Temperature with respect to scaled progress variable for different values of mean mixture fraction. Preliminary results.

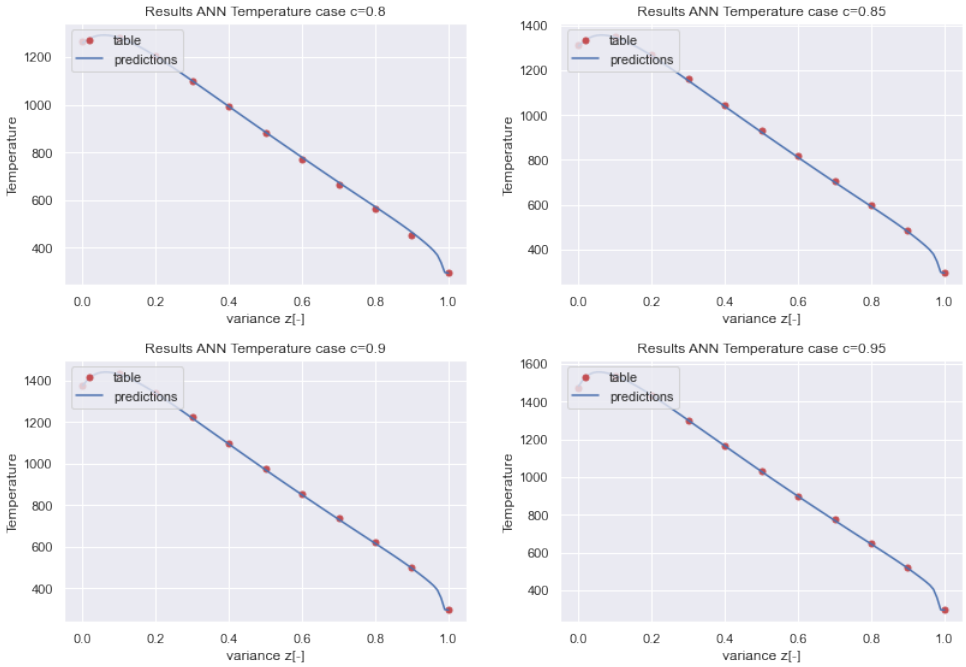


Figure 6-11: Case 2: Temperature with respect to the variance of mixture fraction for different values of mean scaled progress variable. Preliminary results.

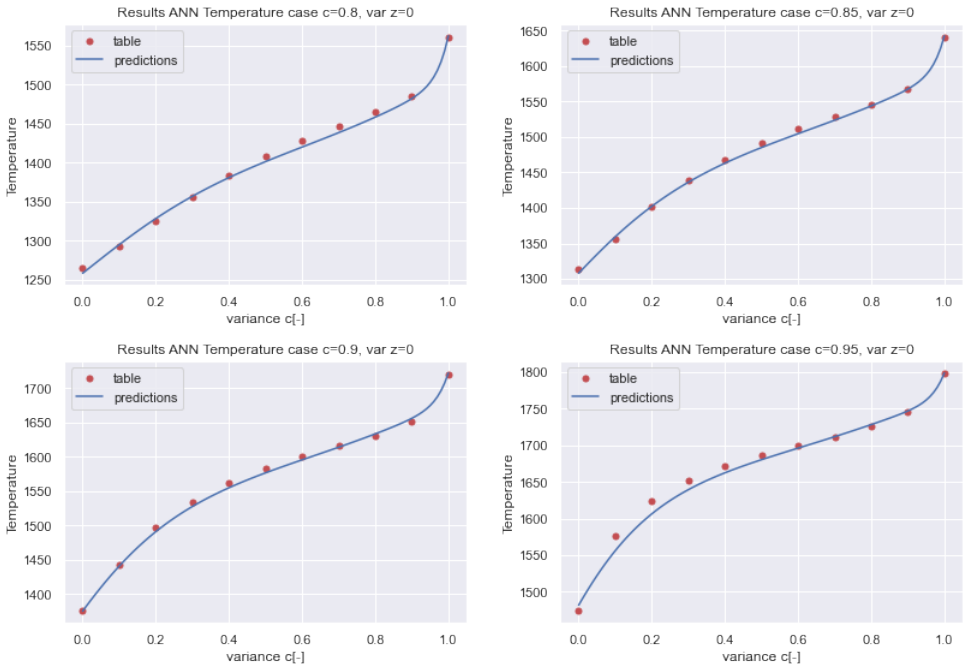


Figure 6-12: Case 3: Temperature with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. Preliminary results.

- $\underline{C_p[\mathrm{J/(kg\ K)}]}$:

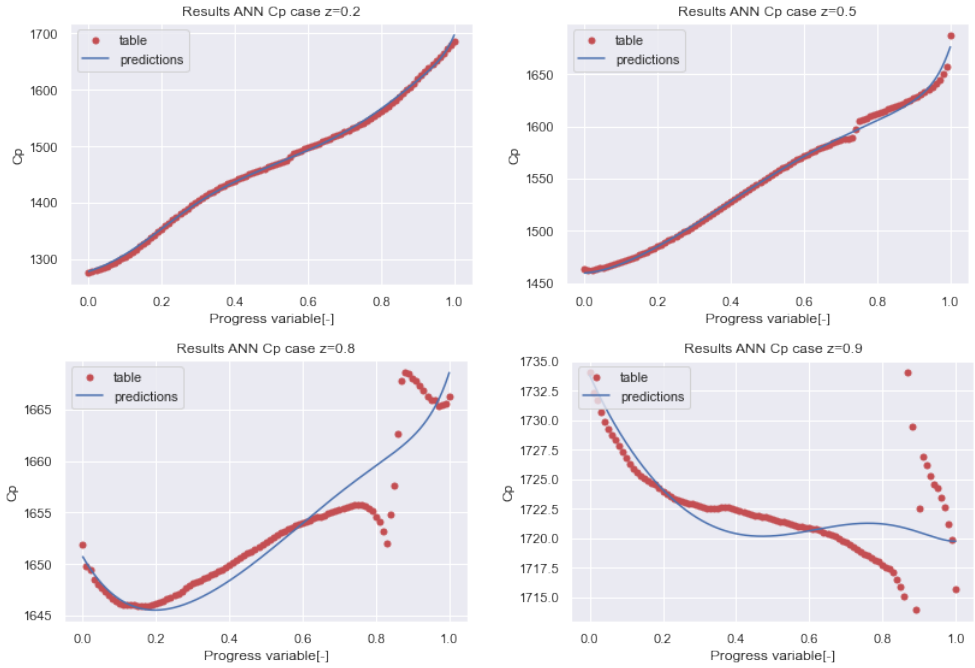


Figure 6-13: Case 1: C_p with respect to scaled progress variable for different values of mean mixture fraction. Preliminary results.

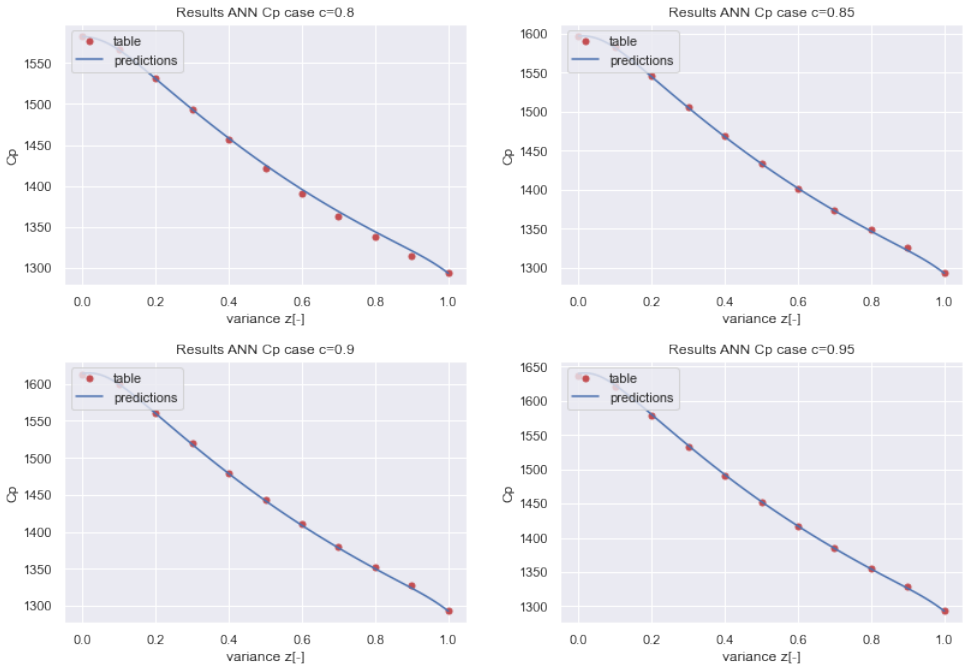


Figure 6-14: Case 2: C_p with respect to the variance of mixture fraction for different values of mean scaled progress variable. Preliminary results.

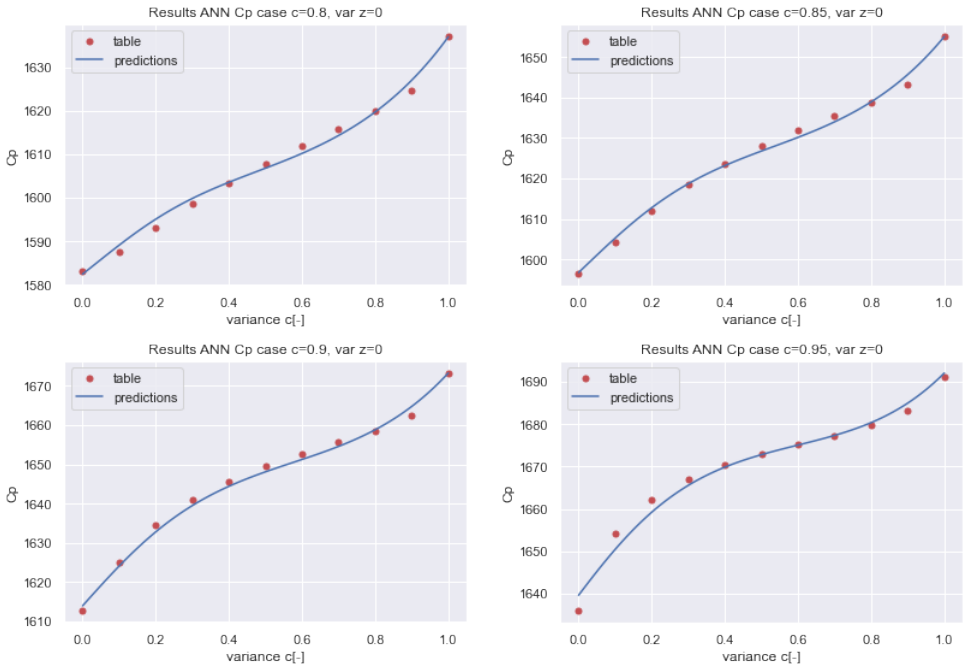


Figure 6-15: Case 3: C_p with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. Preliminary results.

- Mean molar mass of the mixture[g/mol]

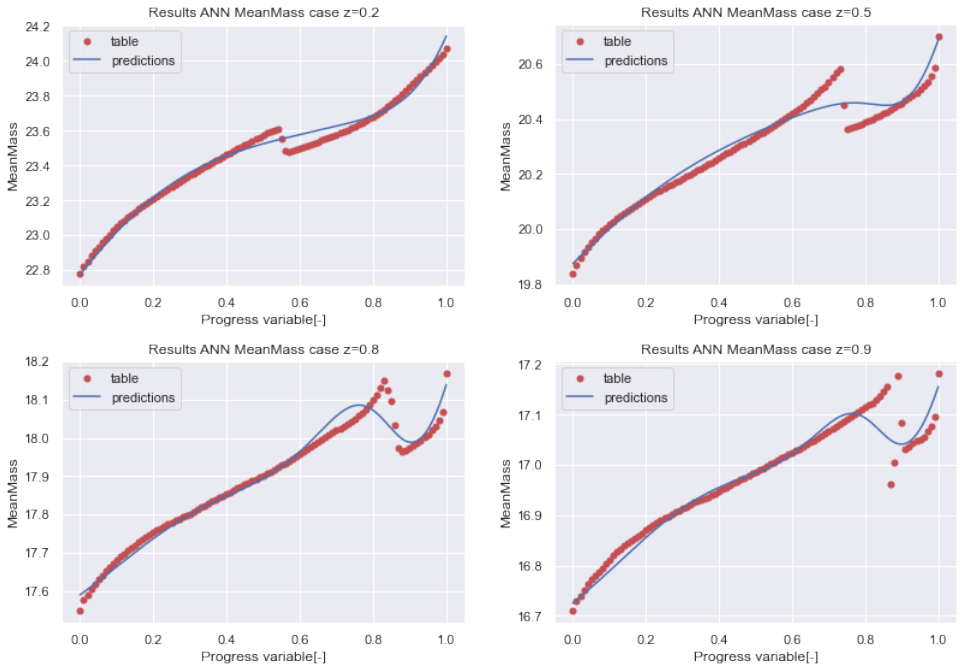


Figure 6-16: Case 1: Mean Molar mass of the mixture with respect to mean scaled progress variable for different values of mean mixture fraction. Preliminary results.

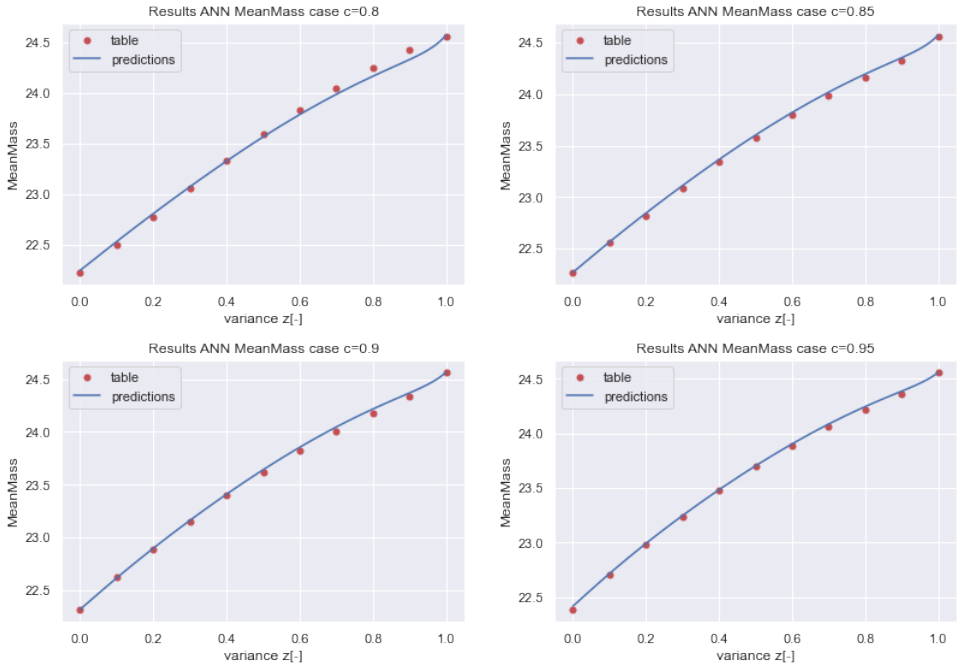


Figure 6-17: Case 2: Mean Molar mass of the mixture with respect to the variance of mixture fraction for different values of mean scaled progress variable. Preliminary results.

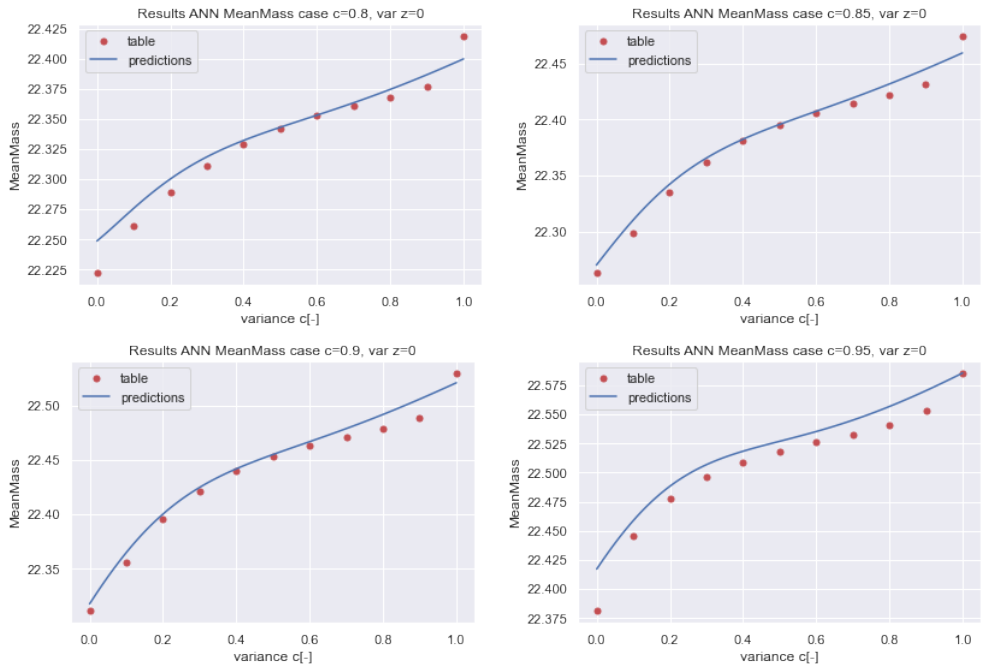


Figure 6-18: Case 3: Mean Molar mass of the mixture with respect to the variance of the scaled progress variable for different values of of mean scaled progress variable. Preliminary Results.

Additionally, we obtain the statistical measures given in the following table 6-2:

Table 6-2: Statistical indicators, training and prediction time, memory requirement for ANN models for different outputs.

Output	t_{train}[s]	t_{pred}[s]	MSE_{train}	MSE_{test}	R^2_{train}	R^2_{test}	memory [kb]
SourcePV	9030.021	0.25346	$1.0128 \cdot 10^{-6}$	$1.0204 \cdot 10^{-6}$	0.99979	0.99979	55
Temperature	6223.883	0.12528	$8.7518 \cdot 10^{-7}$	$8.9343 \cdot 10^{-7}$	0.99995	0.99995	54
C_p	6153.125	0.42465	$1.6589 \cdot 10^{-6}$	$1.6643 \cdot 10^{-6}$	0.99997	0.99997	54
Mean mass	9821.582	0.15941	$9.3244 \cdot 10^{-7}$	$9.2964 \cdot 10^{-7}$	0.99998	0.99998	54

Where t_{train} and t_{pred}, denote the training time and prediction (retrieval) time, respectively. Likewise, MSE_{train}, MSE_{test}, R^2_{train} and R^2_{test} denote the mean squared error(MSE) and coefficient of determination (R^2) of the training (train) and test set, respectively.
From figure 6-7 and 6-10, we can observe that the ANNs are able to capture the underlying physical behaviour of the 4D Lookup table. Furthermore, from figures 6-8,6-9,6-11 and 6-12, we can observe that ANNs are very well capable to represent the behaviour of the source term of the progress variable and temperature over fluctuations in mixture fraction($\widetilde{Z''^2}$) and scaled progress variable ($\widetilde{C''^2}$). This is also confirmed by the statistical measures shown in table 6-2, where the Mean Squared Error (MSE), which compares the values of the 4D lookup table and the predictions obtained using the ANN models are order of 10^{-6} and 10^{-7} in training and test set for the source term of the progress variable and temperature, respectively. The fact that both MSE in training and test set are roughly equal shows that overfitting is not present. This is also verified by the coefficient of determination in both data sets that are approximately

equal and close to 1. This implies that a very accurate fitting has been achieved. Although the flamelet data shows some irregularities at high progress variable in case of rich mixture (high mixture fraction) as it can be seen from figures 6-7 and 6-10. In principle, it would not be so important for applications since this is a very small region in physical space and also because in that range the unscaled progress variable is rarely changing.
Nevertheless, from the results obtained for the outputs, mean molar mass of the mixture and heat capacity, we can observe that there are sharp discontinuities in the case of no fluctuations in mixture fraction and progress variable. These unexpected discontinuities grow for richer mixtures at high progress variable, as it can be seen in figures 6-13 and 6-16. The statistical indicators in table 6-2 do not reflect this nonphysical behaviour in 6-2, where both the MSE and R^2's reflect that the ANNs have done a good job understanding the flamelet data. Therefore, we can conclude that although statistics provides powerful techniques in order to guide us to make a firm conclusion regarding whether a model is accurate, it is always a useful tool to plot the results in order to check effectively if the models are stable or if the data set has some issues as is the case in this project.
In order to figure out where these issues come from, we must come back to the generation of the lookup table. From figures 6-14,6-15,6-17 and 6-15, we can observe that these nonphysical behaviours are not present when fluctuations in mixture fraction and scaled progress variable are considered. Hence, we can conclude that the issues in the lookup table have no relation to the PDF integration. Thereby, the issues may occur in the generation of steady and unsteady flamelets in the FGM approach. This issue will be further elaborated on the next section.

6.1.1.3 Issue 4D Table: flamelets computation

In the FGM approach, a manifold is created by steady flamelets which fill the table for high progress variable and unsteady flamelets which covers the low values of the progress variable (colder zones) where convection and diffusion are simultaneously important. Thus, the steady flamelets are computed for different strain rates from low values (close to zero) until the extinction strain rate. Subsequently, considering the last burning steady flamelet as initial condition, unsteady flamelets at a fixed strain rate set slightly above the extinction strain rate are calculated and reported at different times. In [20], the extinction strain rate considered was $9110[s^{-1}]$, and unsteady flamelets were calculated using a strain rate of $9120[s^{-1}]$. In this way, the following steady and unsteady flamelets shown in figure 6-19 were obtained .

From figure 6-19, we can see that, at first glance, there is no sign of any issue. Nevertheless, as we mentioned in the previous section 6.1.1, the issues presented in some outputs may occur before integrating the fluctuations in mixture fraction and unscaled progress variable. Thus, looking closer at the boundary of the steady and unsteady flamelets (right side figure 6-19), we can observe an overlap at the beginning of unsteady flamelets. This is not expected to occur in the case that the extinction strain rate is chosen correctly. At the right extinction strain rate, the unsteady flamelets at different times must converge to different solutions. Therefore, in the next section, we will depict how to choose the right extinction strain rate, and the new steady and unsteady flamelets for the new extinction strain rate encountered will be shown.

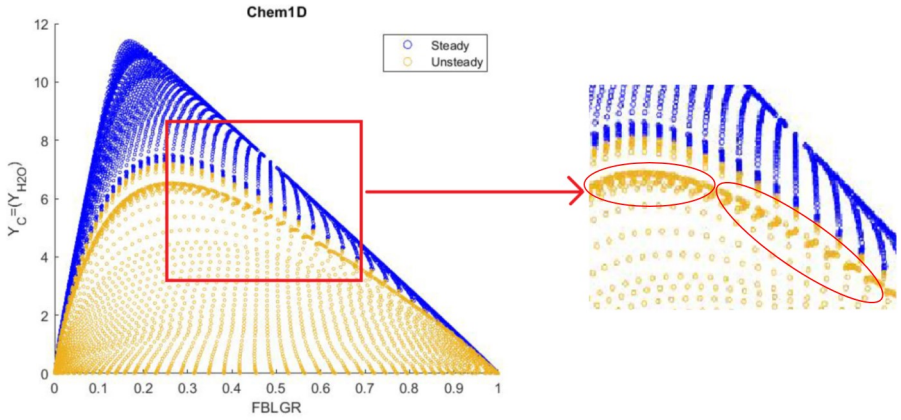


Figure 6-19: Schematic description of issues regarding the generation of steady and unsteady flamelets. Adapted from [20] where Y_C is the progress variable and FBLGR is the Bilger's mixture fraction.

6.1.2 Final Results

In this section, we will present the final results once the issue presented in the 4D FGM lookup tables has been fixed.

6.1.2.1 Solution 4D table

As it was explained in the previous section 6.1.1.3, the extinction strain rate may not have been chosen correctly in [20]. In this way, as the strain used in [20] was $9110[s^{-}]$, we take this as previous knowledge and we proceed as follows: we recompute the steady flamelet but considering an extinction strain rate far from the previous extinction strain rate, in this case $9300[s^{-1}]$. This higher strain rate is based on the following assumption: for strain rates smaller than the extinction strain rate, the solution should converge at few numerical iterations. However, when the strain rate is higher than the extinction strain rate, the numerical computations should take much more time because the solution should not converge even considering a very high number of iterations. Thus, we compute steady flamelets at every increment of $10[s^{-1}]$ of the strain rate in the range $9000[s^{-1}]$ to $9300[s^{-1}]$. While the computations were carried out, it was encountered that at $9190[s^{-1}]$ the steady flamelet was not converging even though the maximum number of iterations was reached. Therefore, the previous strain rate $9180[s^{-1}]$ should correspond to the extinction strain rate for this hydrogen flame.
Thus, we compute steady flamelets from a strain rate of $10[s^{-1}]$ until the extinction strain rate of $9180[s^{-1}]$ and unsteady flamelets at strain rate $9190[s^{-1}]$ considering the last burning steady flamelet(at strain rate $9180[s^{-1}]$) as initial condition for the unsteady flamelets. In this way, 136 and 144 steady and unsteady flamelets are computed, respectively, and a new 4D FGM table is created where steady and unsteady flamelets are plotted in 6-20.

Comparing 6-19 and 6-20, we can observe that both show a peak of the progress variable at approximately $Z = 0.2$ and have the same shape. However the problem related to the boundary between steady and unsteady flamelets has disappeared(right side figure 6-20).

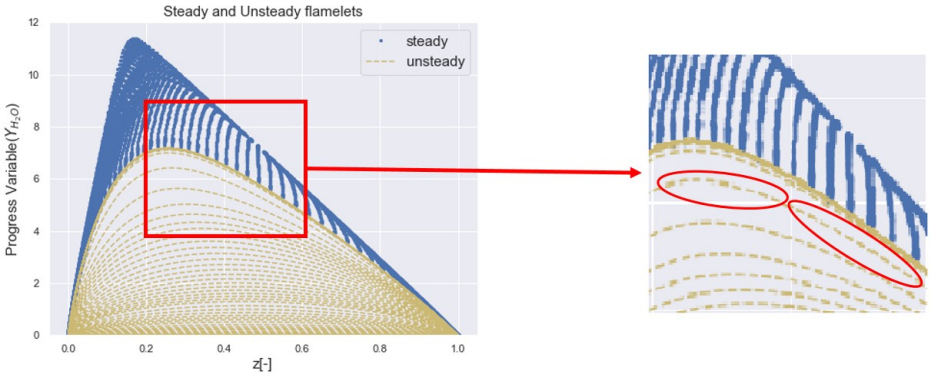


Figure 6-20: New steady and unsteady flamelets generated using a new extinction strain rate of $9180[s^{-1}]$.

Additionally, we compare the new table with the experimental data reported in [19], which is shown in figure 6-21.

From figure 6-21, we can observe that the new 4D lookup table is in agreement with the

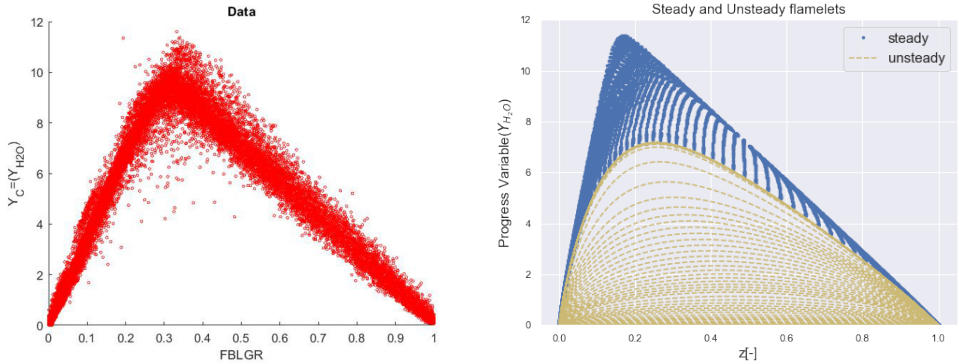


Figure 6-21: Comparison between experimental data(left) and the new 4D lookup table(right).(experimental data adapted from [20])

shape and the maximum of the experimental data. Nevertheless the peak is predicted around $Z = 0.2$ instead of $Z = 0.31$ shown in the experimental data. Although the experiment also includes the effects of differential diffusion, this difference related to the peak position may be because flamelets are sufficiently disturbed by the flow to supress the effects of differential diffusion. This must be understood in future research.

Hence, we can conclude that a better choice of the extinction strain rate in the calculations of steady and unsteady flamelets has fixed the unexpected issues encountered in this project. In the following section, we will train the ANN models again with the new 4D lookup table. Furthermore, we will see how these models show better performance.

6.1.2.2 ANN models for different outputs

Using the new 4D Lookup table, we train the Artificial Neural Network (ANN)s keeping the architecture and procedure used previously(see sections 6.1.1 and 5.4). Thus, considering the aforementioned cases 1-2-3 explained in section 6.1.1.1, we obtain the following results:

• Source Progress Variable[kg/(m^{3}s)]:

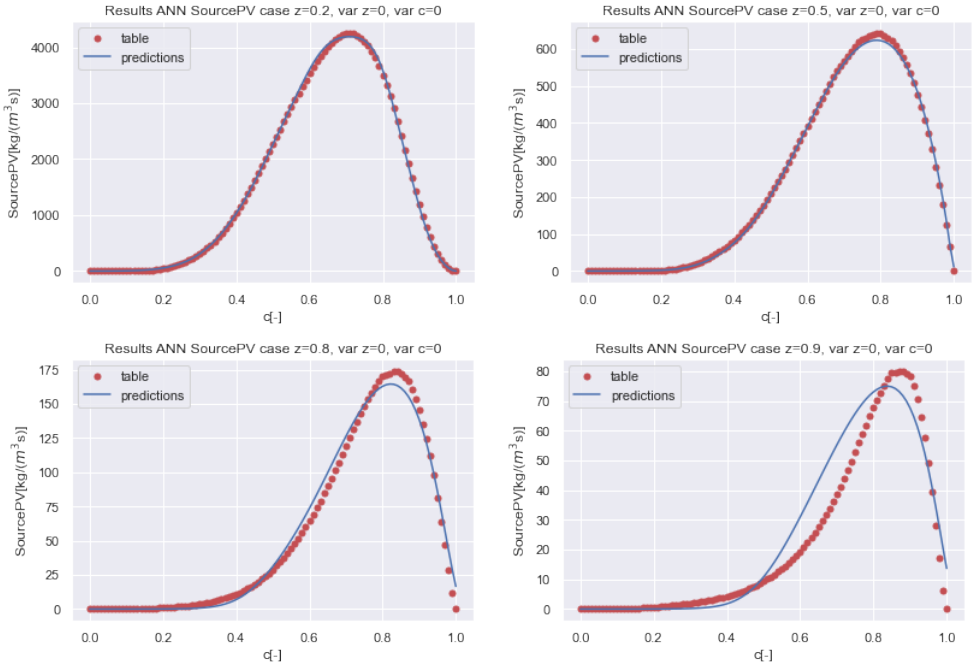


Figure 6-22: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction. Final results.

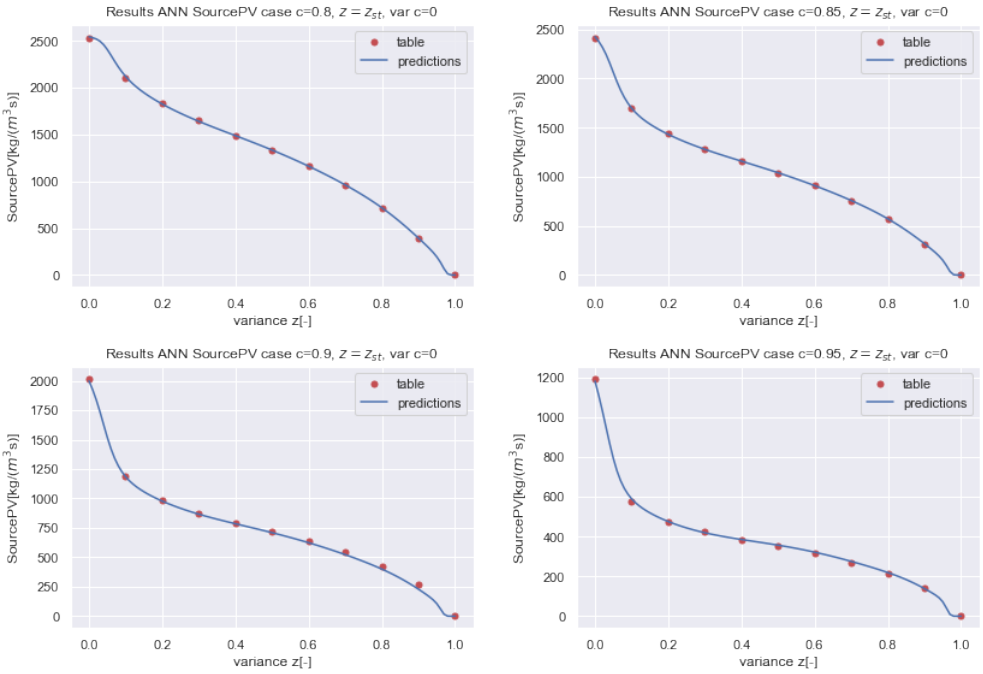


Figure 6-23: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable. Final results.

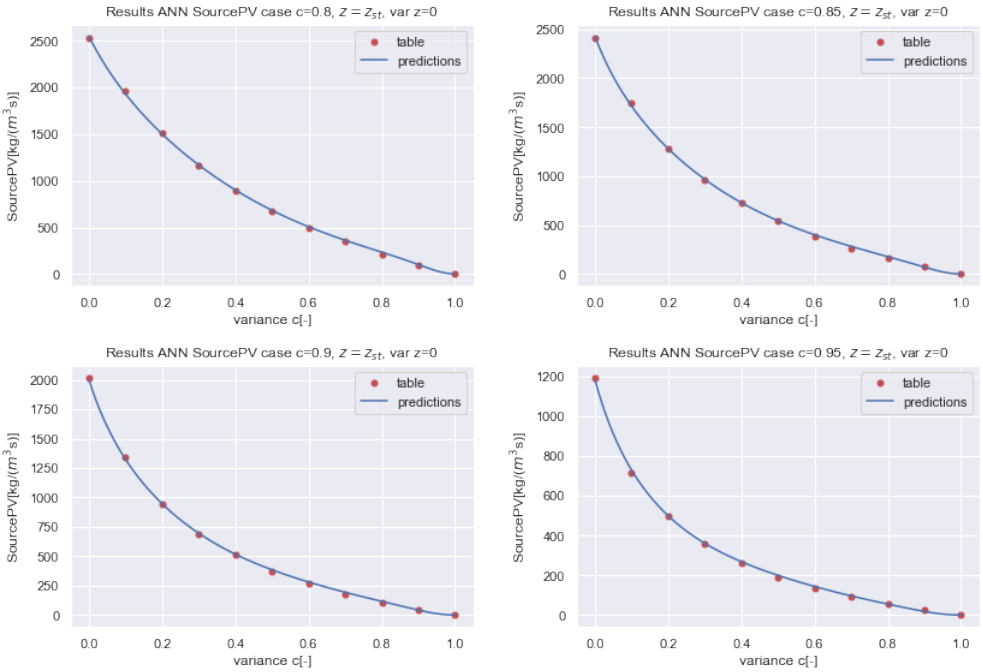


Figure 6-24: Case 3: Source term of the progress variable with respect to the variance of progress variable for different values of mean scaled progress variable. Final results.

- Temperature[K]:

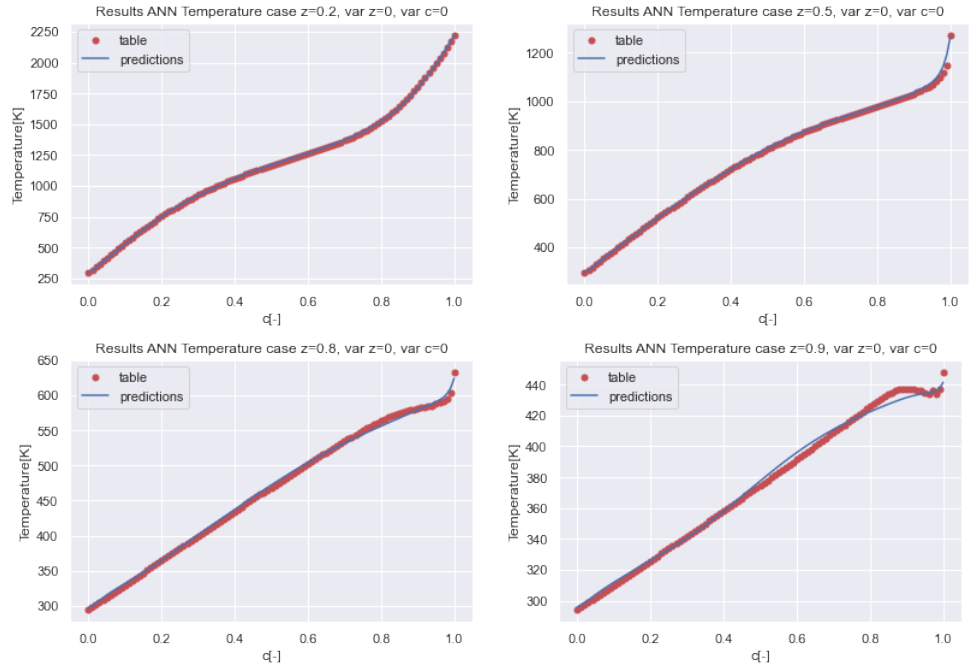


Figure 6-25: Case 1: Temperature with respect to mean scaled progress variable for different values of mean mixture fraction. Final results.

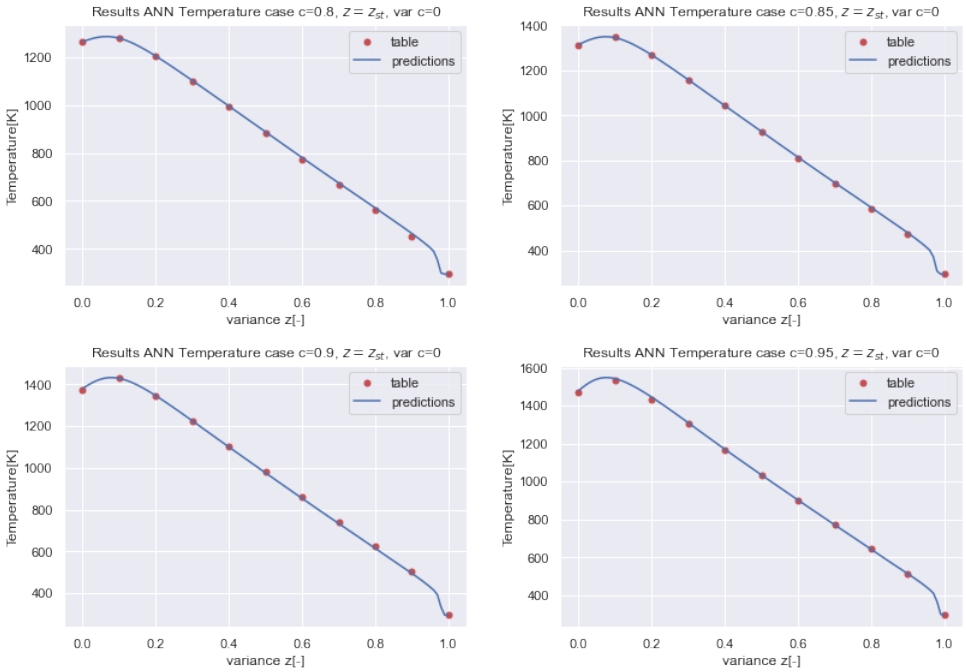


Figure 6-26: Case 2: Temperature with respect to variance of mixture fraction for different values of mean scaled progress variable. Final results.

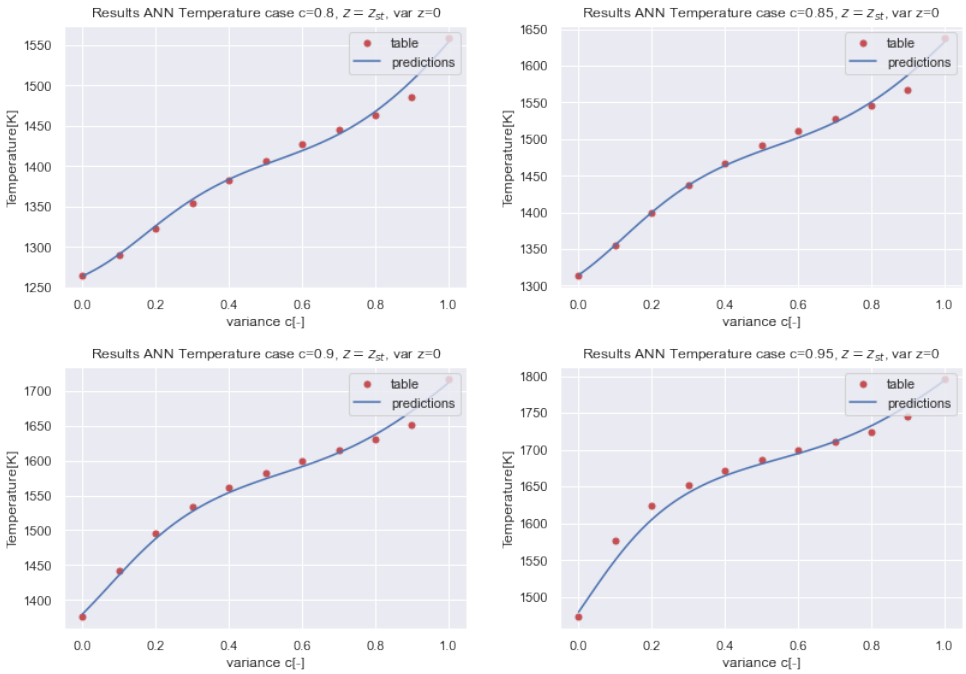


Figure 6-27: Case 3: Temperature with respect to variance of the scaled progress variable for different values of mean scaled progress variable. Final results.

- $C_p[\mathrm{J}/(\mathrm{kg\ K})]$:

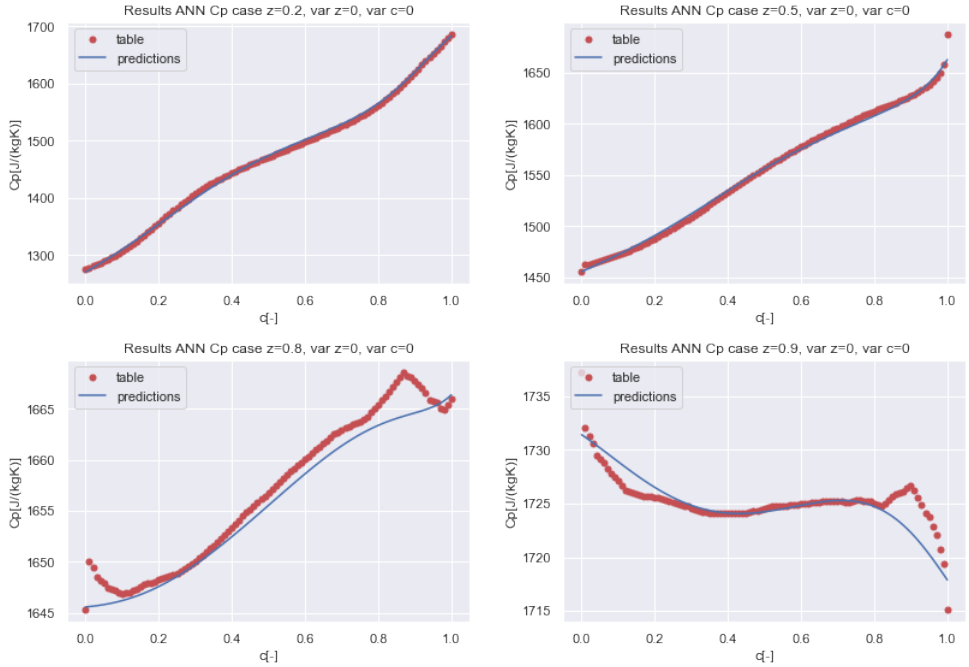


Figure 6-28: Case 1: C_p with respect to mean scaled progress variable for different values of mean mixture fraction. Final results.

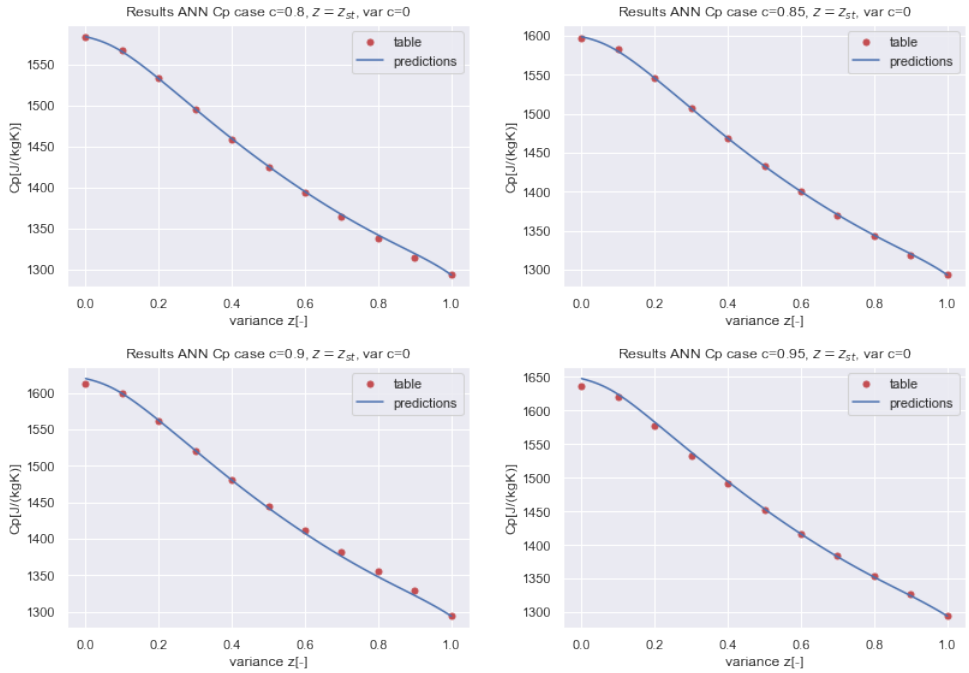


Figure 6-29: Case 2: C_p with respect to variance of mixture fraction for different values of mean scaled progress variable. Final results.

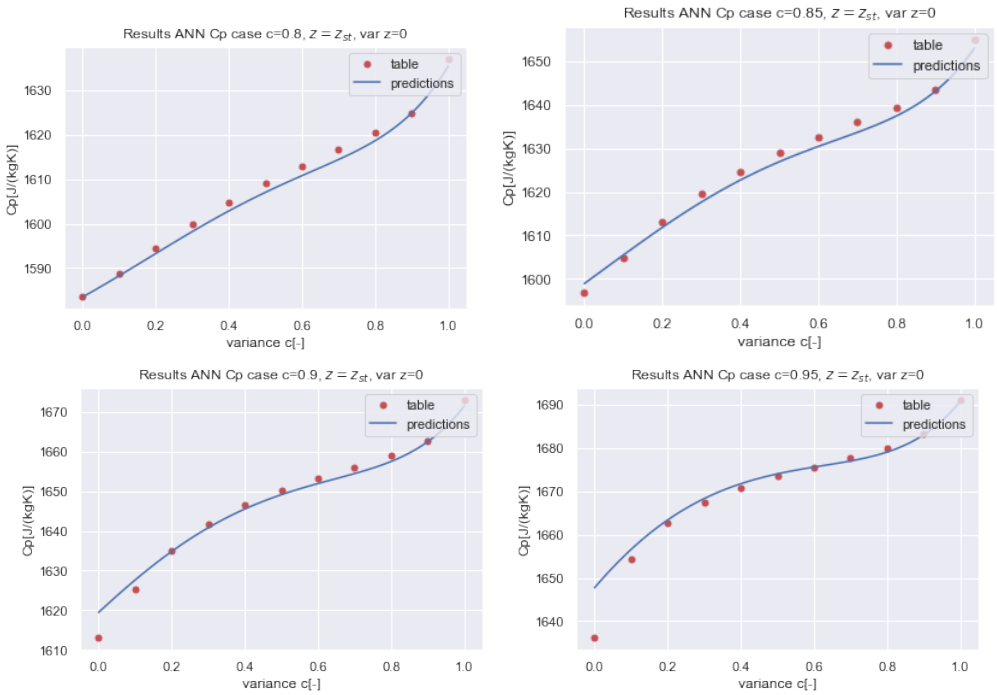


Figure 6-30: Case 3: C_p with respect to variance of the scaled progress variable for different values of mean scaled progress variable. Final results.

- Mean molar mass of the mixture[g/mol]

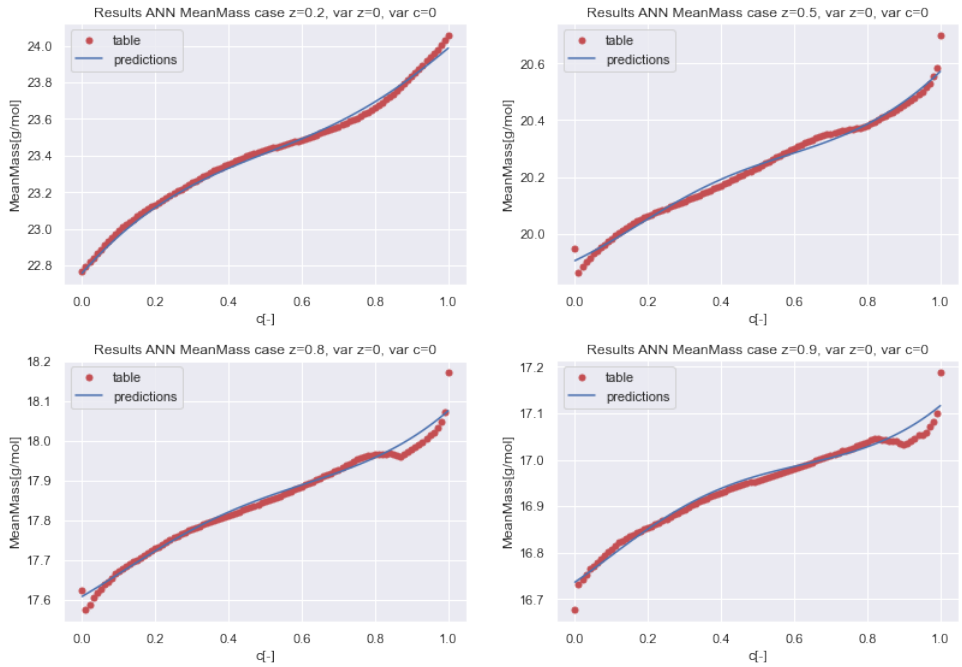


Figure 6-31: Case 1: Mean Molar mass of the mixture with respect to mean scaled progress variable for different values of mean mixture fraction. Final results.

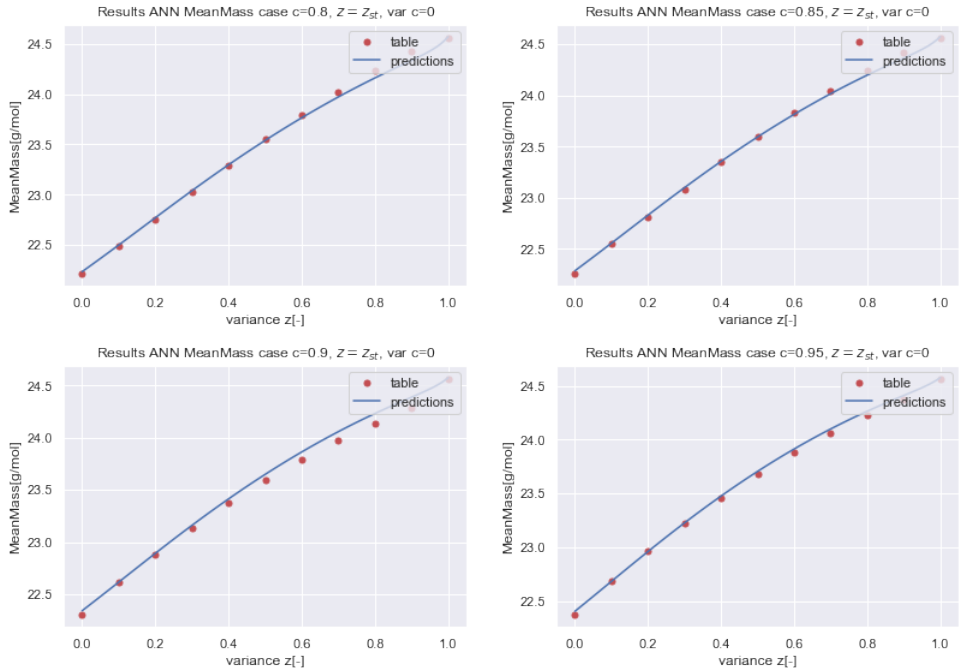


Figure 6-32: Case 2: Mean Molar mass of the mixture with respect to variance of mixture fraction for different values of mean scaled progress variable. Final results.

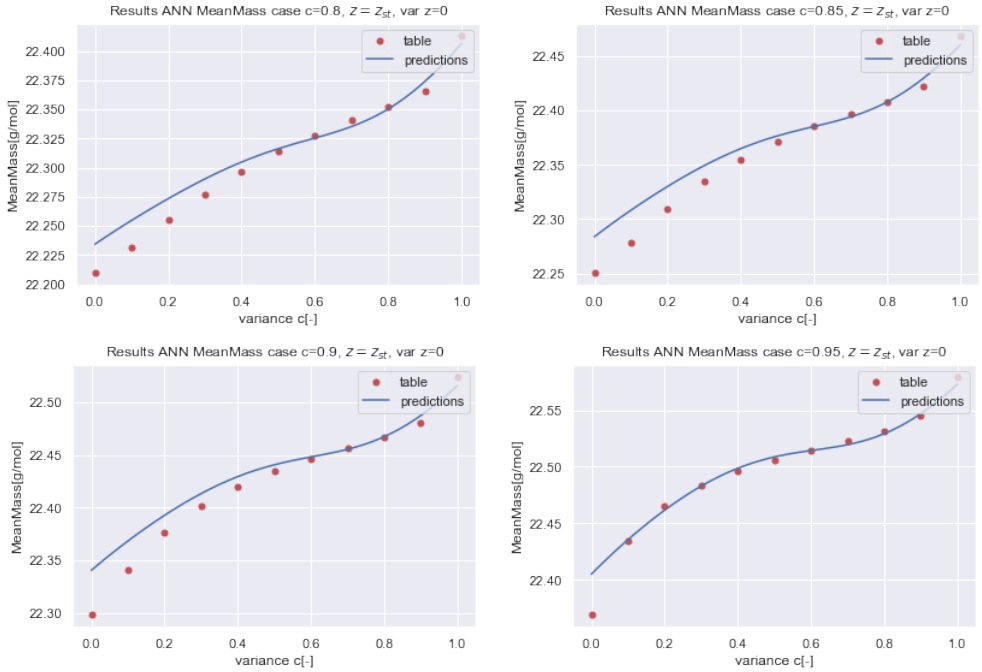


Figure 6-33: Case 3: Mean Molar mass of the mixture with respect to variance of the scaled progress variable for different values of mean scaled progress variable. Final Results.

The statistical measures and training and prediction time are summarised in the following

table 6-3:

Table 6-3: Statistical indicators, training and prediction time, memory requirement for ANN models for different outputs of the new 4D lookup table.

Output	MSE_{train}	MSE_{test}	R^2_{train}	R^2_{test}	t_{train}[s]	t_{pred}[s]
C_p	$1.24428{\cdot}10^{-6}$	$1.23859{\cdot}10^{-6}$	0.999980	0.999980	9683.374	0.050987
Density	$2.90148{\cdot}10^{-6}$	$2.92510{\cdot}10^{-6}$	0.999928	0.999928	9300.986	0.067274
H_2O_2	$1.89499{\cdot}10^{-5}$	$2.39758{\cdot}10^{-5}$	0.998569	0.998193	7534.424	0.15825
H_2O	$2.84964{\cdot}10^{-7}$	$2.81689{\cdot}10^{-7}$	0.999988	0.999989	7061.082	0.12892
H_2	$6.64536{\cdot}10^{-7}$	$6.66804{\cdot}10^{-7}$	0.999991	0.999991	6678.862	0.062501
HO_2	$5.71510{\cdot}10^{-5}$	$5.50952{\cdot}10^{-5}$	0.997193	0.997295	2919.26	0.124999
H	$6.21589{\cdot}10^{-6}$	$6.08647{\cdot}10^{-6}$	0.999521	0.999531	6388.448	0.162355
Mean Mass	$8.33418{\cdot}10^{-7}$	$8.27341{\cdot}10^{-7}$	0.999988	0.999988	10058.70	0.071956
N_2	$4.94074{\cdot}10^{-7}$	$5.10637{\cdot}10^{-7}$	0.999994	0.999994	5152.653	0.117927
O_2	$2.49793{\cdot}10^{-7}$	$2.47302{\cdot}10^{-7}$	0.999997	0.999997	4931.536	0.092883
OH	$2.73983{\cdot}10^{-6}$	$2.68629{\cdot}10^{-6}$	0.999020	0.999061	5201.218	0.124088
O	$2.72881{\cdot}10^{-6}$	$2.89858{\cdot}10^{-6}$	0.998567	0.998472	5011.494	0.0625
SourcePV	$1.87124{\cdot}10^{-6}$	$1.81386{\cdot}10^{-6}$	0.999617	0.999628	6011.434	0.123022
Temperature	$1.88581{\cdot}10^{-6}$	$1.90657{\cdot}10^{-6}$	0.999902	0.999902	8647.162	0.207926

From figure 6-22, we can see that at high mixture fractions, the issues mentioned in section 6.1.1 have been solved. Additionally, from figures 6-23 and 6-24, we can observe that the ANN is well capable to represent the flamelet data for turbulent flames. The statistical measures also reflect this in table 6-3, where the error is order 10^{-6} in both training and test set. Likewise, the coefficient of determination are 0.999617 and 0.999628 in training and test data set, respectively, which implies that a very accurate fitting has been achieved. Although for richer mixture (bottom plots in figure 6-22) it seems at first glance that the ANN has a worse performance in that area, it must be noted that the vertical scale is smaller in comparison to the other plots in the same figure (from 0 to 80 in comparison to the other scales that are in the range 0 to 4000). So, it cannot be concluded that the ANN performs better or worse in some areas. In fact, we can conclude that the ANN model has a good performance in each area of the table.

Similar conclusions can be made in the case of temperature. From figure 6-25, we can observe that the ANN makes an excellent job capturing the higher temperatures closer to the stoichiometric state. This can also be observed where fluctuations in mixture and progress variable are considered (figures 6-26 and 6-27). From the bottom right plot at figure 6-25, we can observe that at high unscaled progress variable, the temperature is not strictly increasing. This could be a consequence of particular chemical reactions that may be occurring in that area of the physical space. This behaviour is also present in the 6D lookup table as will be explained in the section 6.2.1. From this particular trend, we can conclude that temperature cannot depict a combustion process unambiguously, in other words, temperature should not be used as a progress variable.

On the other hand, figures 6-28 and 6-31 show that the nonphysical behaviour has been almost completely solved. The new results show a good match between steady and unsteady flamelets. However, we can observe in figure 6-28 that at $\widetilde{C} = 0$, the heat capacity has a discontinuity which becomes larger at high values of mixture fraction. This could be a con-

sequence of a round off error while the flamelets were generated, which may induce the error in C_p. This must be understood in future research.
Regarding the ANNs for heat capacity, it seems from figure 6-28 that C_p might be very difficult to represent by ANN. However, from the same figure, we can see that where it seems that ANN has a poor performance, the scale is very small (the range is 5). Thus, the error made in rich mixtures is very small. This can be confirmed by the statistical behaviour for the C_p (table 6-3), where the error is order 10^{-6} and the coefficient of determination is 0.999980 in both training and test sets. Therefore, as it was concluded for the source term of the progress variable, the ANN performs quite good along the whole area in the 4D lookup table.
Furthermore, from table 6-3, we can conclude that the ANNs are able to represent each output with high accuracy . Additionally, we can observe different training times for thermochemical quantities that have the same ANN architecture. This difference is explained as a consequence of using Early Stopping technique.
Finally, in the following table 6-4, we compare the memory requirement of a lookup table (single output) with respect to the memory requirement of the ANN models obtained:

Table 6-4: Comparison of memory requirements between 4D lookup table(Single output) with ANN models obtained in the current project.

Table size	Memory requirement
4D Lookup table (one output)	26.5MB
ANN	$54 - 55$kB

From table 6-4, we can observe a substantial memory reduction of around 500 times. This memory saving could be used for performing more detailed numerical simulation in parallel computing mode using more cores. Although it was not possible to perform a numerical simulation in a CFD solver during the current project, a larger retrieval time may be expected using ANNs in comparison to retrieving the information from the lookup table. This is expected due to the simple fact that an ANN is a much more complex model than the usual linear interpolation used for retrieving the information from the lookup tables. Nevertheless, the majority of time in CFD simulation is usually spent on solving the transport equations rather than retrieving information from the lookup tables[11]. Therefore the expected increase in retrieving information using ANNs may not increase substantially the overall time usually spent on CFD simulations.

6.2 6D Diluted Air FGM (DA-FGM) Dutch natural gas table

Using the knowledge gained on training ANNs for the 4D lookup table, we proceed to extend the use of ANNs for the 6D lookup table generated using the DA-FGM approach for the case of dutch natural gas. Then, we follow the procedure explained in section 5.4, finding out that the most suitable ANN architectures are: 30-30-30-30 for the source term of the progress variable and 26-26-26-26 neurons for the rest of the thermochemical quantities stored in the lookup table. As this table extends the Flamelet Generated Manifold (FGM) approach to diluted and Non-adiabatic flames, we consider the three cases mentioned in section 6.1.1.1 and added a fourth case including heat loss. These cases are described as follows:

- Case 1: Case without fluctuations, adiabatic and no dilution - the output varies with respect to the mean scaled progress variable $\widetilde{C}$, for different values of mean mixture fraction $\widetilde{Z}$, considering $\widetilde{C''^2} = \widetilde{Z''^2} = \widetilde{\eta} = \widetilde{\gamma} = 0$.
- Case 2: Stoichiometric mixtures with fluctuations in mixture fraction, without heat losses and no dilution - the output varies with respect to the variance of mixture fraction $\widetilde{Z''^2}$ at stoichiometric mixture fraction $\widetilde{Z} = \widetilde{Z_{st}}$, for different and high values of the mean scaled progress variable $\widetilde{C}$ and considering $\widetilde{C''^2} = \widetilde{\eta} = \widetilde{\gamma} = 0$.
- Case 3: Stoichiometric mixtures with fluctuation in the scaled progress variable without heat loss and no dilution - the output varies with respect to the variance of scaled progress variable $\widetilde{C''^2}$ at stoichiometric mixture fraction $\widetilde{Z} = \widetilde{Z_{st}}$, for different and high values of the mean scaled progress variable $\widetilde{C}$ and considering $\widetilde{Z''^2} = \widetilde{\eta} = \widetilde{\gamma} = 0$.
- Case 4: Diluted stoichiometric mixtures with heat losses without fluctuations in mixture fraction and scaled progress variable - the output varies with respect to the mean enthalpy loss factor $\widetilde{\eta}$ at stoichiometric mixture fraction $\widetilde{Z} = \widetilde{Z_{st}}$ and at a given mean dilution level $\widetilde{\gamma}$, for different and high values of the mean scaled progress variable $\widetilde{C}$ and considering $\widetilde{Z''^2} = \widetilde{C''^2} = 0$.

6.2.1 Results

First, we show the results obtained for the most difficult output to be represented by an ANN which is the source term of the progress variable. The results are shown as follows:
• Source term of the progress Variable[kg/(m^{3}s)]:

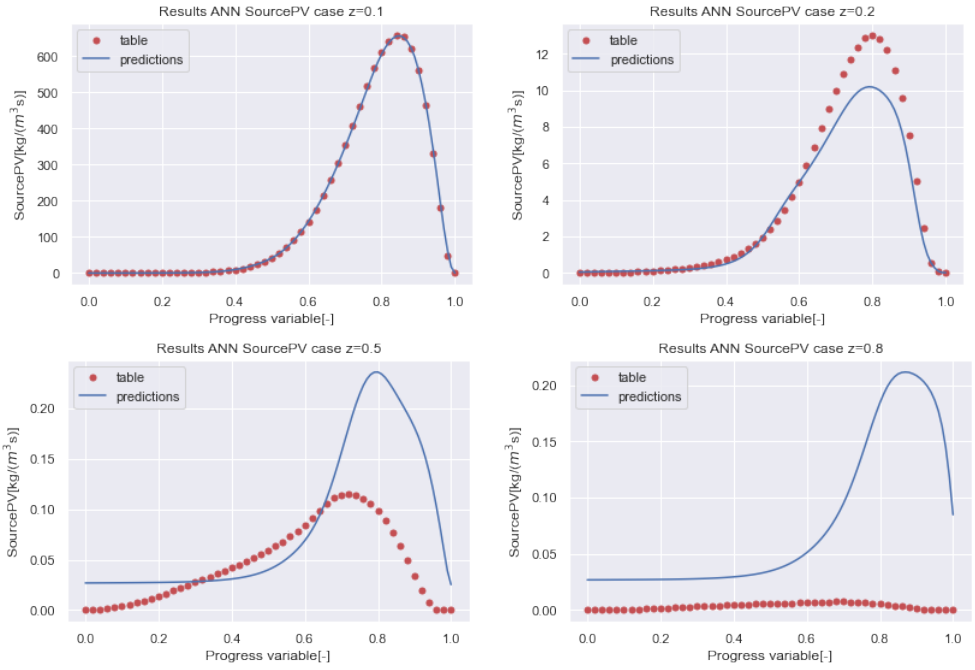


Figure 6-34: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table.

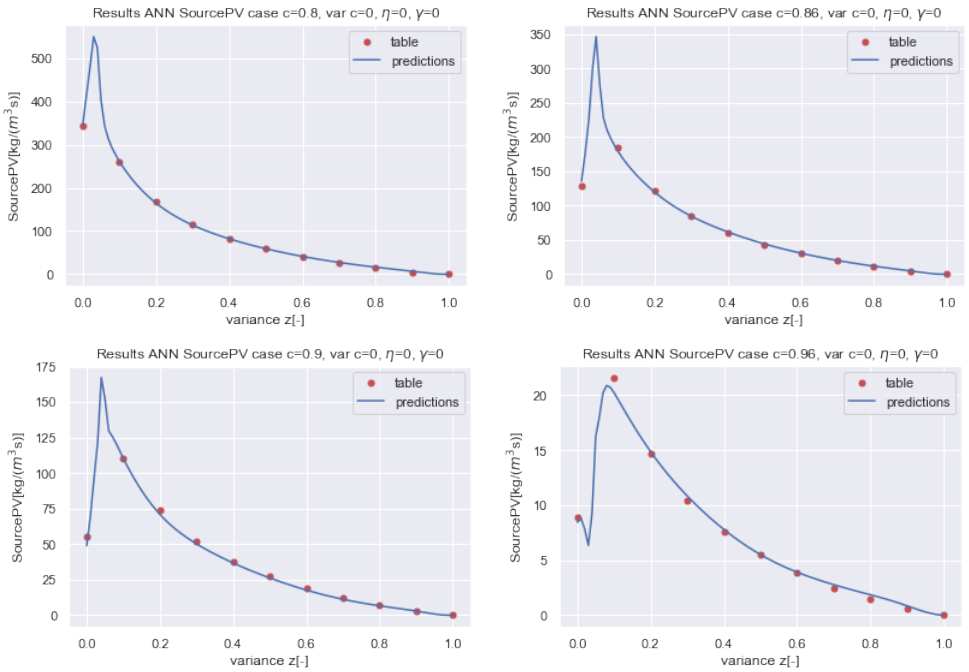


Figure 6-35: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table.

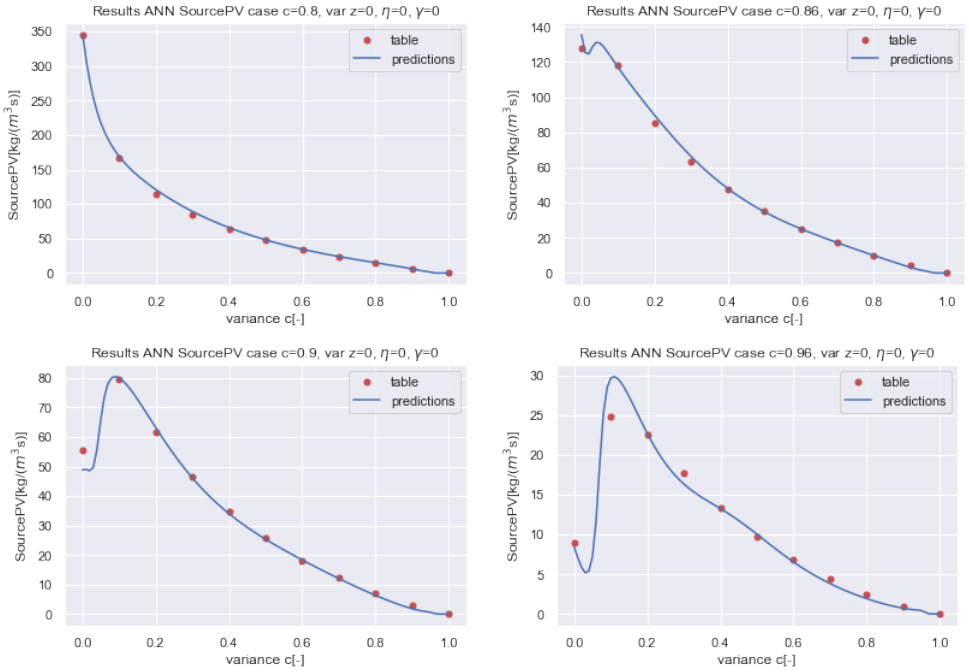


Figure 6-36: Case 3: Source term of the progress variable with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. 6D Table.

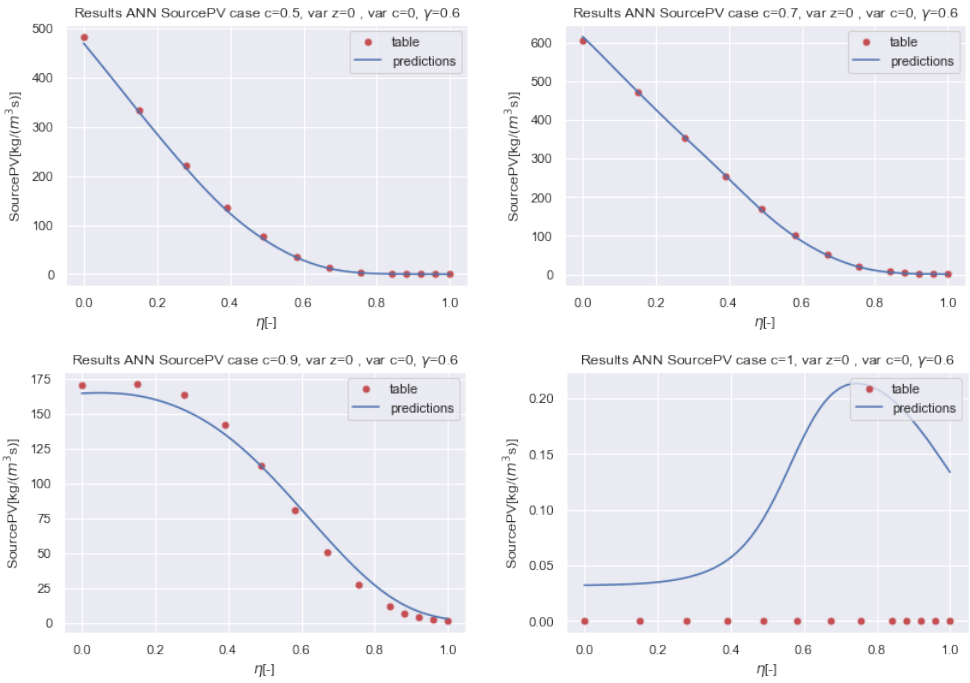


Figure 6-37: Case 4: Source term of the progress variable with respect to enthalpy loss factor for different values of the scaled progress variable. 6D table.

From figure 6-34, we can observe that the source term of the progress variable reaches the higher values for lean mixtures. Additionally, we can observe that the ANN is able to capture the peak of this output successfully. Moreover, the model is stable in the case of no fluctuations in mixture fraction and progress variable. Although, when the mixture fraction is higher, we can observe some deviations between the prediction given by the ANN and the value stored in the table (plots at the top right, bottom left and right in figure 6-34), we must note the following: for $\widetilde{\xi} = 0.2$, the largest deviations between the prediction made by the ANN and the value store in the lookup table is around 2.5 when $\widetilde{C} \approx 0.8$. The same observation can be made for richer mixtures when the deviations between the predictions and the true values are smaller than 0.25. The ANN may have been made similar errors for lean mixtures but due to the large scale of the source term of the progress variable in those cases(intervals of hundreds), these errors cannot be observed in the plot. Therefore, we can conclude that the ANN have the same accuracy and stability along laminar-adiabatic and non-diluted flamelets. Similar conclusions can be made in the case 3 and 4 from figures 6-36 and 6-37, respectively, where the ANN is well capable to represent the fluctuations in progress variable and when diluted-non adiabatic effects are taking into account. Although some wiggles are developed for higher values of the progress variable (figure 6-36) at small fluctuations in the progress variable (closer to zero), they are very small in size and disappear almost immediately when fluctuation in the progress variable increases. Therefore, we can conclude that the model can represent fluctuations in progress variable successfully and when heat losses and dilution are considered.

However, in case 2 of fluctuation in mixture fraction (figure 6-35), we can observe nonphysical peaks for smaller fluctuation in mixture fraction (closer to zero). Although they are confined in a narrow area closer to zero (no fluctuations), these peaks are large (order of hundreds)

for $\widetilde{C} = 0.8$ and $\widetilde{C} = 0.86$, and becomes smaller until they disappear for higher values of $\widetilde{C}$. In order to study whether these nonphysical peaks are present along the table, we plot case 2 but considering non-adiabatic($\widetilde{\eta} \approx 0.5$) and diluted flamelets ($\widetilde{\gamma} = 0.5$). Thus, we obtain the results shown in figure 6-38:

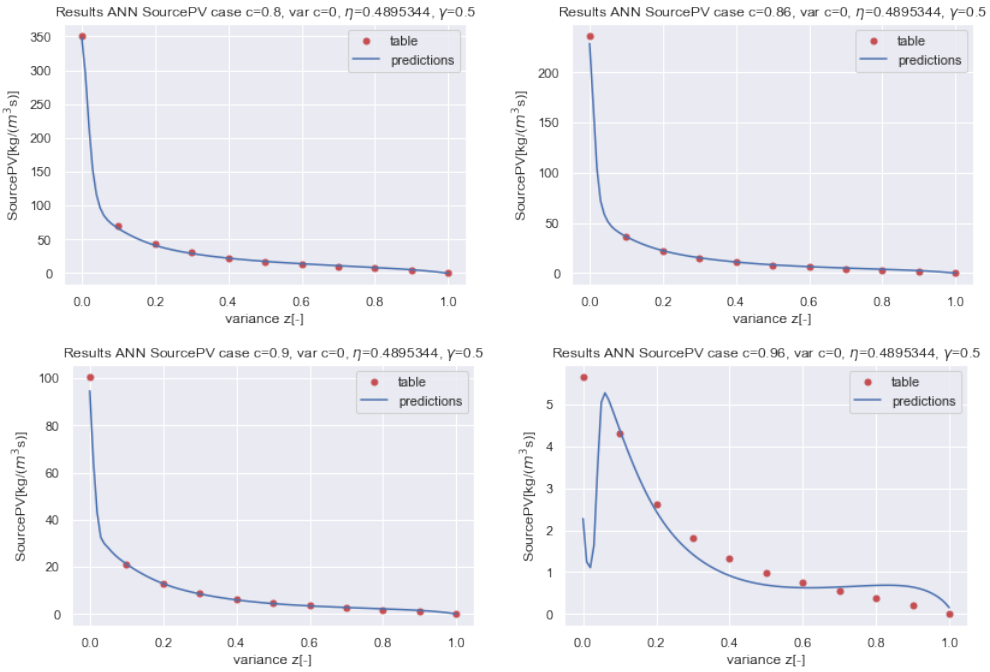


Figure 6-38: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean progress variable considering $\widetilde{\gamma} = 0.5$ and $\widetilde{\eta} \approx 0.5$. 6D Table.

In figure 6-38, the sharp peaks at low variance as observed in figure 6-35 are absent. Although fo the case $\widetilde{C} = 0.96$ a wiggle is developed, its magnitude is very small (less than 4 in size). Hence, we can conclude that non-physical sharp peaks are localized closer to non diluted-adiabatic flames and when fluctuations in mixture fraction are small (closer to zero). In order to address this issues, other ANN architectures are considered: 1) increasing the number of neurons per hidden layer to 36 and 2) adding an additionally hidden layer and reducing the neurons per layer to 16. The first option is based on the assumption that increasing the number of neurons would allow the model fitting better the data and, at the same time, forcing the model to lower the undesired peaks. The second option is based on the assumption that adding an additional hidden layer would allow mapping the data into another high dimensional space where the regression problem would be, in principle, easier to achieve. In order to avoid large amount of figures, the results for these architectures can be found in the appendix A of this report. Contrary to our expectation, these two strategies cause the opposite effect: Non-physical peaks become larger and more wiggles are developed. As possible explanation, it can be pointed out that these sharp peaks present in the ANN arise in areas where the ANN cannot compare its predictions with the true values of the table. As it can be seen in figure 6-35, these peaks appear in the interval of fluctuations in mixture fraction of $[0, 0.1]$ and the table only contains the values of the source term of the progress variable at 0 and 0.1. Therefore, the model is unable to know whether its predictions are accurate within this range.

Hence, we can conclude that a refinement of the mesh in the variance of mixture fraction in the range $[0, 0.1]$ would be required in order to inform the ANN about the true values in that area and, indirectly, force the model to eliminate the non-physical peaks present in that range.

On the other hand, for the rest of the thermochemical quantities stored in the 6D lookup table, we here show only the results of temperature, kinematic viscosity and grey absorption coefficient. Moreover, the following table 6-5 summarizes the statistical measures for the most important outputs in the 6D lookup table:

Table 6-5: Statistical indicators, training and prediction time, memory requirement for ANN models for different outputs of the 6D lookup table.

Output	MSE_{train}	MSE_{test}	R^2_{train}	R^2_{test}	t_{train}[s]	t_{pred}[s]
alpha	$2.29116{\cdot}10^{-7}$	$2.29437{\cdot}10^{-7}$	0.999994	0.999994	31687.63	0.09248
$\overline{kI_b}$	$9.24757{\cdot}10^{-7}$	$9.25337{\cdot}10^{-7}$	0.999971	0.999971	38922.24	0.061967
$\overline{k}$	$6.20130{\cdot}10^{-7}$	$6.21295{\cdot}10^{-7}$	0.999982	0.999982	31136.22	0.043862
psi	$2.43773{\cdot}10^{-7}$	$2.43958{\cdot}10^{-7}$	0.999995	0.999995	40089.48	0.161509
SourcePV	$1.86031{\cdot}10^{-7}$	$1.84412{\cdot}10^{-7}$	0.999412	0.999417	20890.64	0.099103
Temperature	$4.98109{\cdot}10^{-7}$	$4.98189{\cdot}10^{-7}$	0.999987	0.999987	15904.38	0.192252
variance T	$3.20249{\cdot}10^{-7}$	$3.19699{\cdot}10^{-7}$	0.999966	0.999966	29918.38	0.084579
Viscosity	$1.64439{\cdot}10^{-6}$	$1.64529{\cdot}10^{-6}$	0.999961	0.999960	34596.42	0.084856
$\overline{Y_c \dot{w}_{Y_C}}$	$1.41469{\cdot}10^{-7}$	$1.41901{\cdot}10^{-7}$	0.999475	0.999476	34236.36	0.043884

- Temperature[K]:

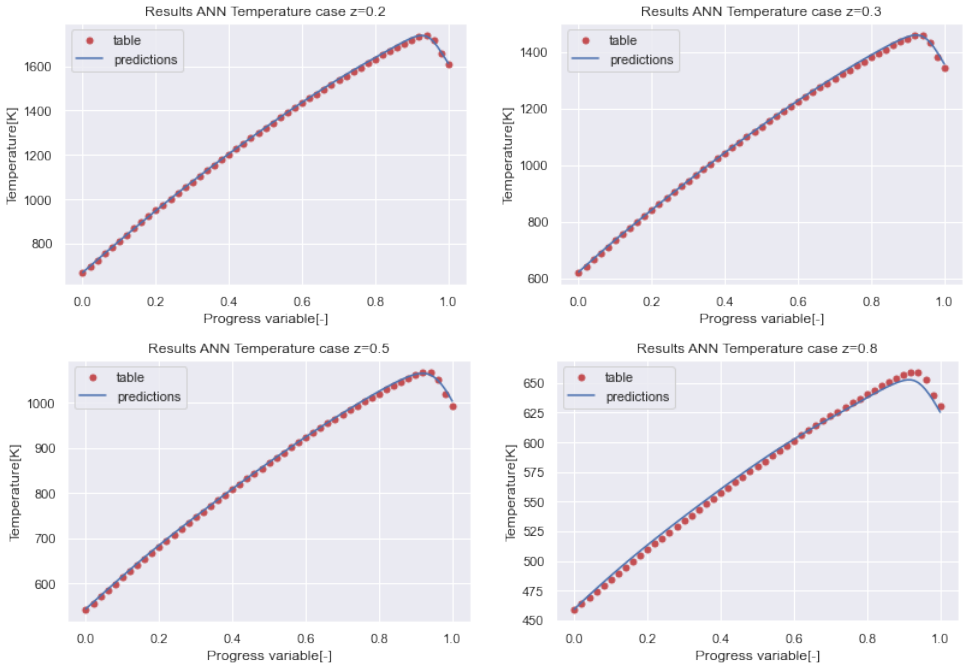


Figure 6-39: Case 1: Temperature with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table.

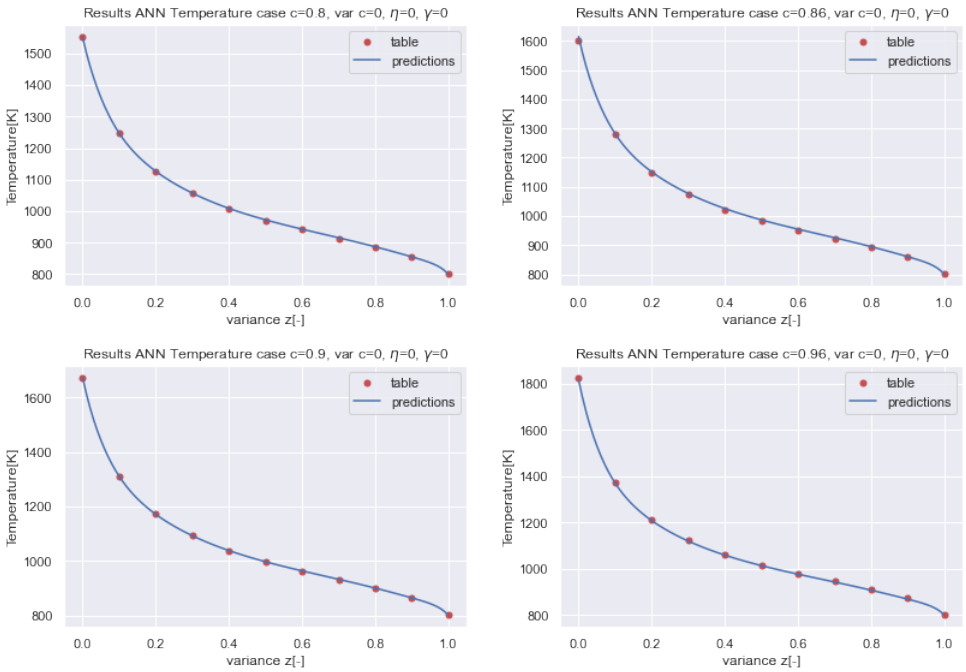


Figure 6-40: Case 2: Temperature with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table.

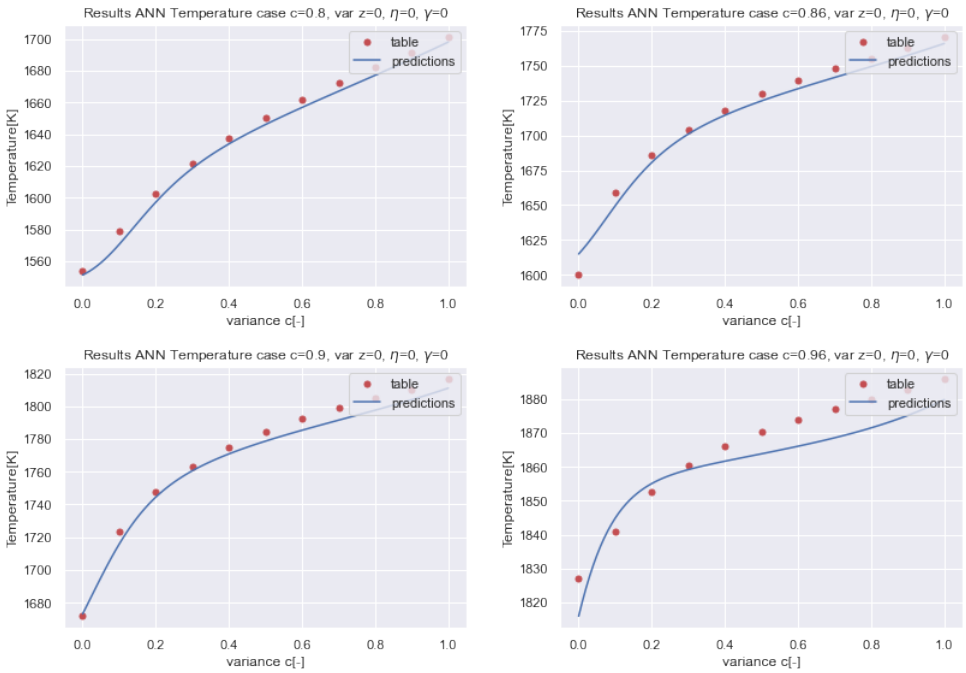


Figure 6-41: Case 3: Temperature with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. 6D Table.

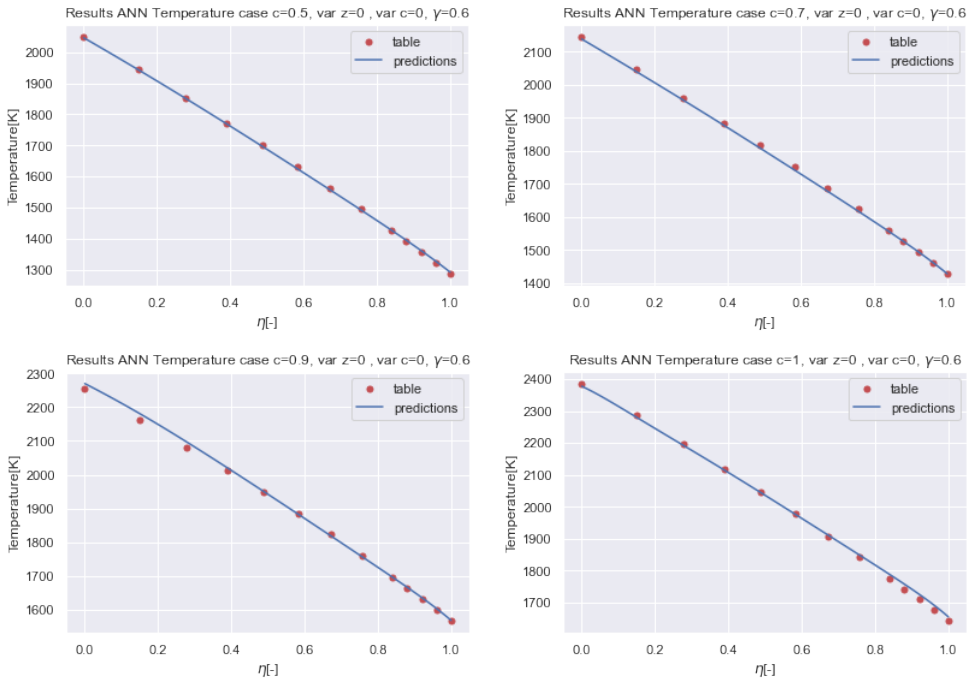


Figure 6-42: Case 4: Temperature with respect to enthalpy loss factor for different values of the mean scaled progress variable. 6D table.

- Kinematic viscosity $\nu[\mathrm{m}^2/\mathrm{s}]$:

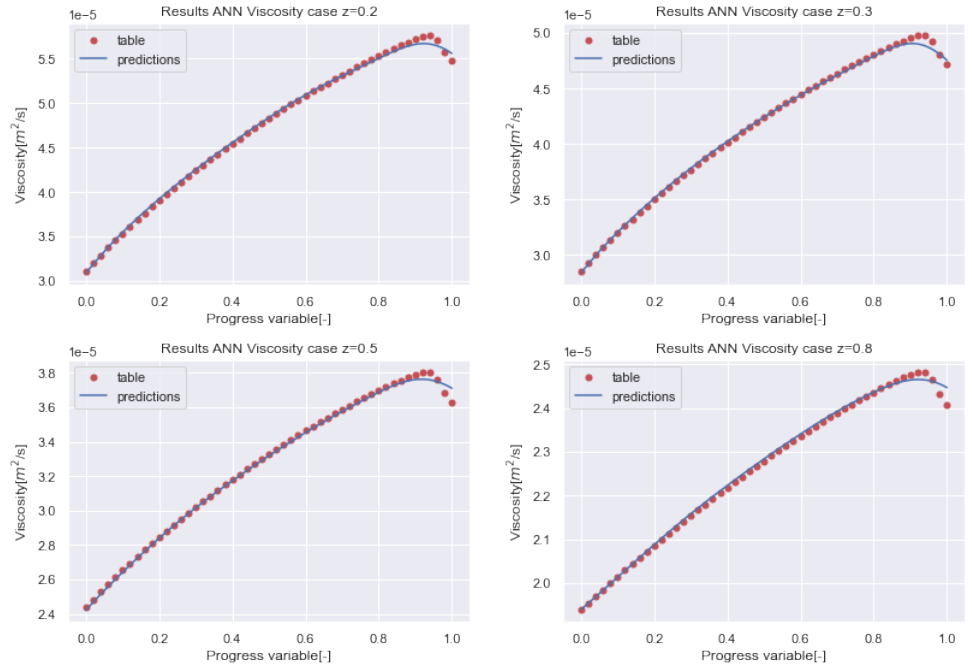


Figure 6-43: Case 1: Viscosity with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table.

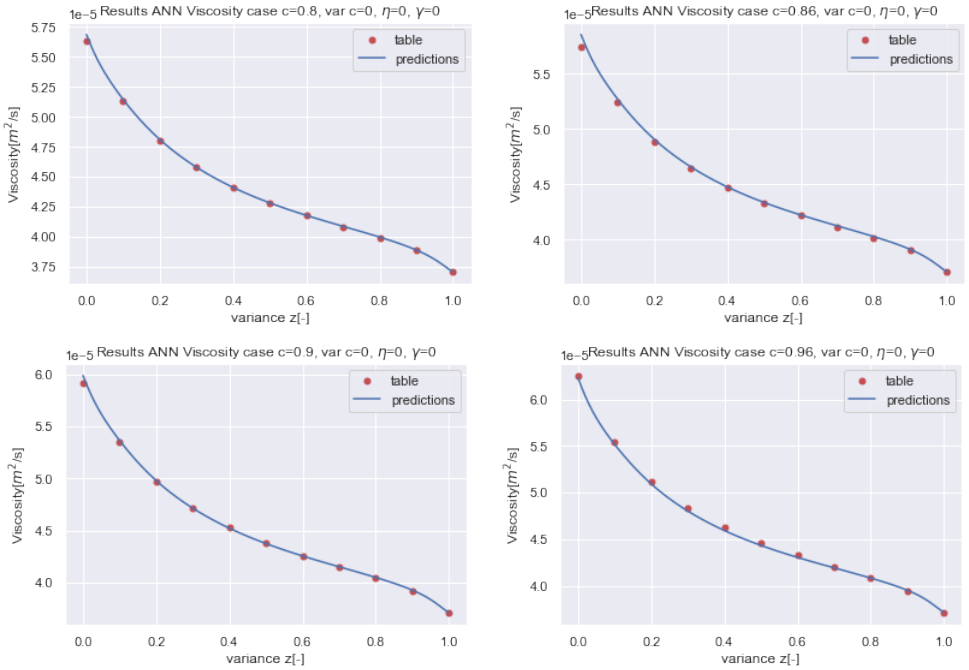


Figure 6-44: Case 2: Viscosity with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table.

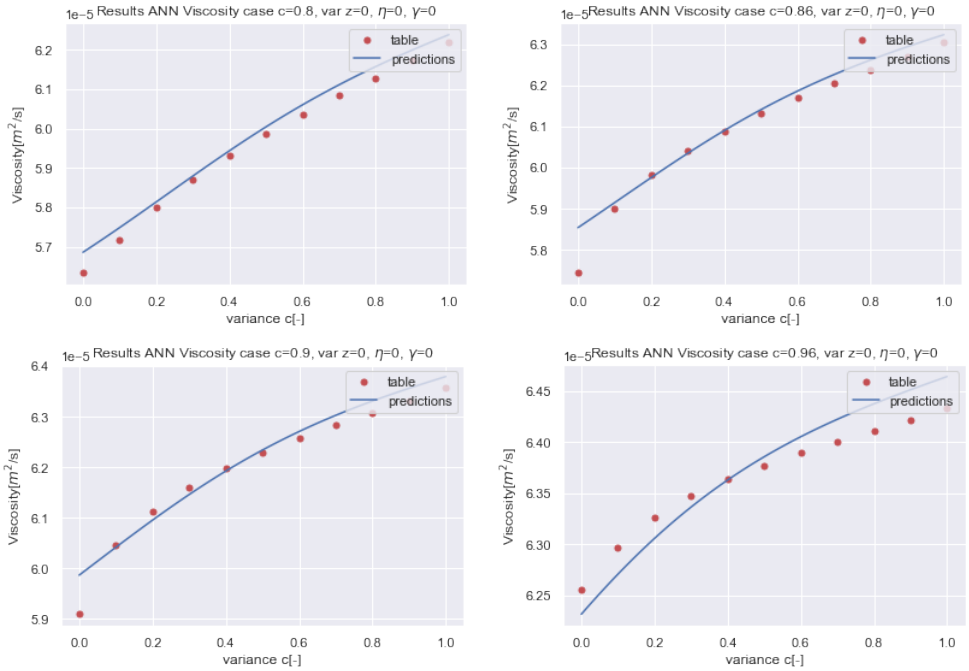


Figure 6-45: Case 3: Viscosity with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. 6D Table.

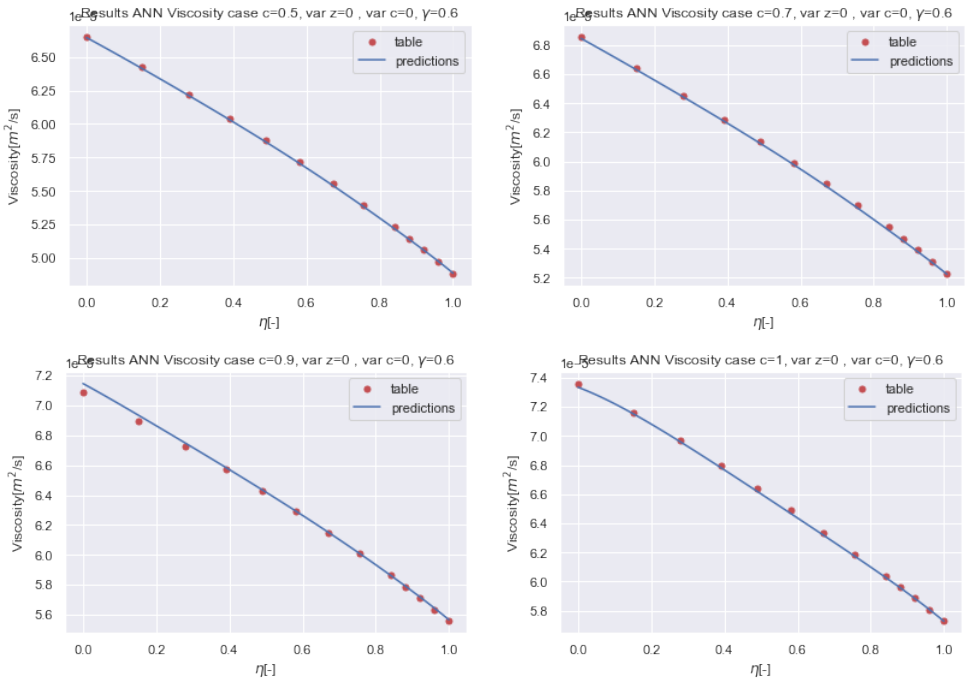


Figure 6-46: Case 4: Viscosity with respect to enthalpy loss factor for different values of the mean scaled progress variable. 6D table.

- Grey absorption coefficient $\kappa[\mathrm{m}^{-1}]$:

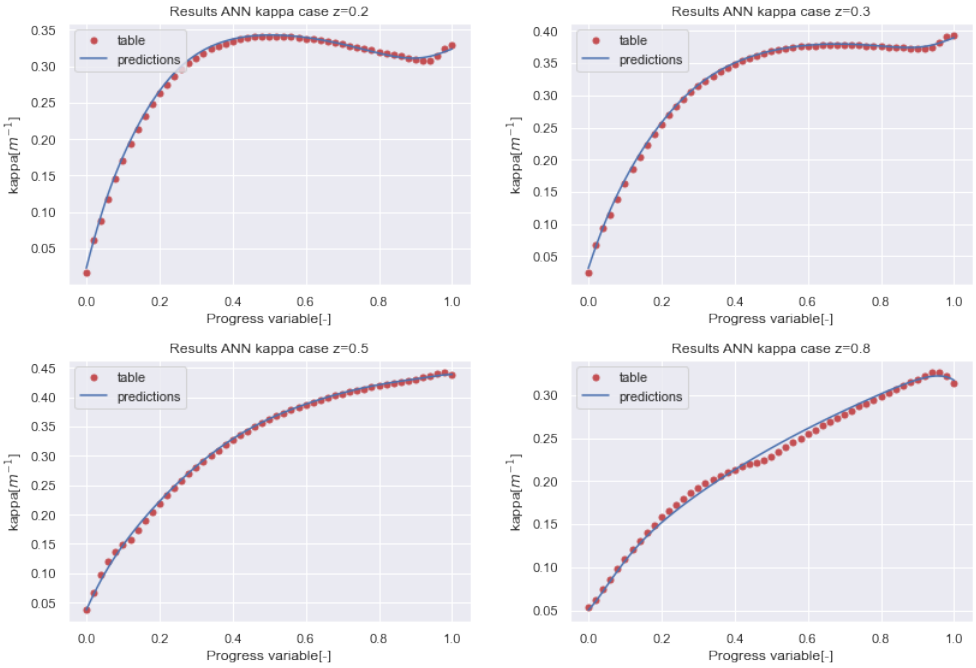


Figure 6-47: Case 1: Grey absorption coefficient with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table.

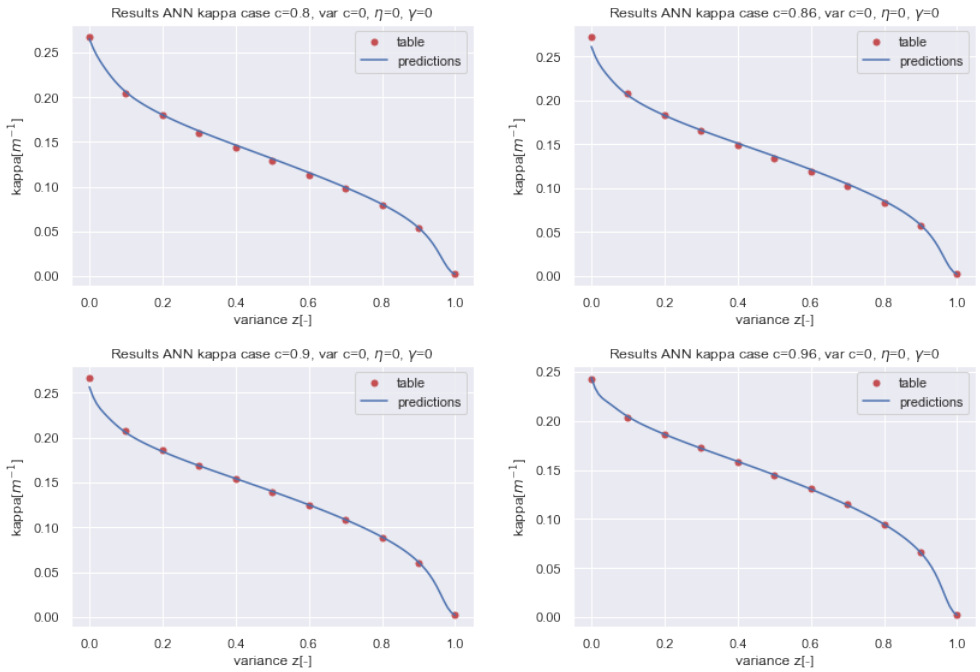


Figure 6-48: Case 2: Grey absorption coefficient with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table.

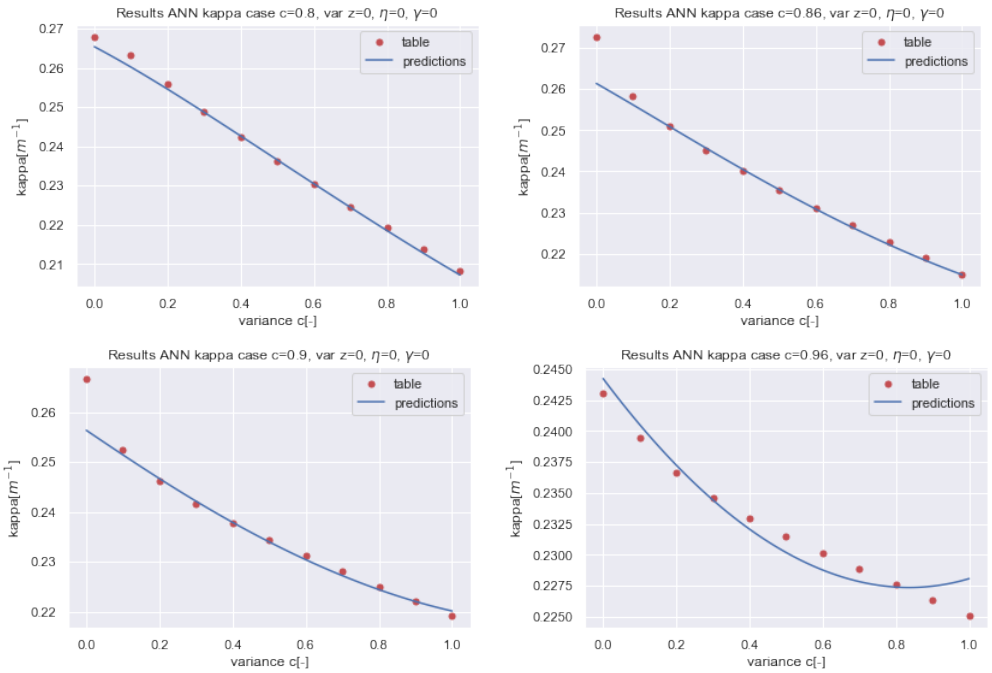


Figure 6-49: Case 3: Grey absorption coefficient with respect to the variance of the scaled progress variable for different values of mean scaled progress variable. 6D Table.

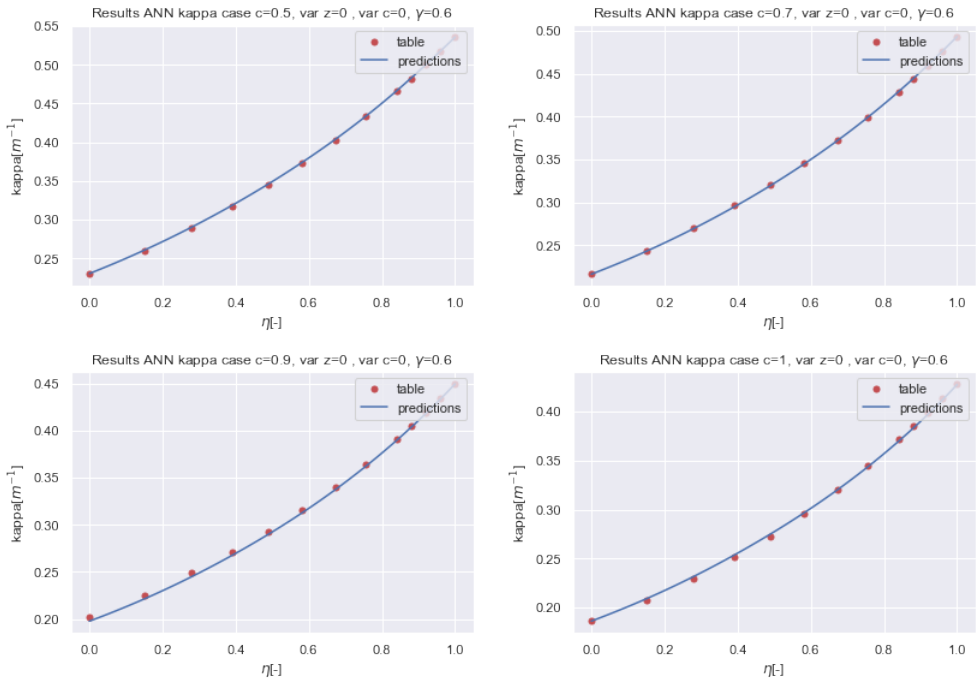


Figure 6-50: Case 4: Grey absorption coefficient with respect to enthalpy loss factor for different values of the mean scaled progress variable. 6D table.

From the above figures, we can observe that, in general, the ANN models are very well capable to represent the outputs along the whole table. This can be confirmed observing the statistical measures in table 6-5, where the MSE in training and data set are around the same and order 10^{-7} in both cases. Likewise, the coefficient of determinations in both sets are very close to 1, which implies that accurate fitting has been achieved. This can also be seen from figures 6-39 to 6-50. Regarding to these outputs, we can see that their scales do not change along the table, for instance, in the case of temperature, the scale is order 10^2, whereas in kinematic viscosity is order 10^{-5}. Conversely, in the source term of the progress variable, there is a wider range of scales from 0 to 10^2. This fact combined with the high non-linearity of this output may explain why it is so difficult to represent it using an ANN in comparison to the other outputs.
Moreover, as it was pointed out in section 6.1.2, we can observe from figure 6-39 that the temperature is not strictly increasing. Therefore, we can conclude that temperature does not define unambiguously a state during combustion and it should not be used as a progress variable.
Finally, in the following table 6-6, we compare the memory requirement of a lookup table (single output) with respect to the memory requirement of the ANN models obtained:

Table 6-6: Comparison of memory requirements between 6D lookup table(Single output) with ANN models obtained in the current project.

Table size	Memory requirement
6D Lookup table (one output)	625.3MB
ANN	72 – 80kB

From table 6-6, we can observe a substantial memory reduction of around 7800 times. Therefore, a substantial reduction in memory has been achieved. Additionally, with these models performing Large Eddy Simulations (LES) may be possible and more reliable results could be obtained. Although it was not possible to perform a numerical simulation in a CFD solver during the current project, the retrieval time is expected to be larger than retrieving information from the lookup table. Nonetheless, this retrieval time is expected to be more similar to retrieving the information using lookup tables than was the case in the 4D table. This is because, as the 6D lookup tables contains 45 millions points, searching information in those tables may be quite similar to retrieving information from ANNs. Therefore, retrieving information using ANNs in the case of the 6D lookup table eventually will not cause an increase on the overall time during CFD simulations.

Chapter 7

Conclusions and Recommendations

In this master thesis project, Artificial Neural Networks have been applied in order to simplify and replace the chemistry representation provided by lookup tables in turbulent combustion. From the results, we can conclude the following:

- Using Artificial Intelligence, in this case Artificial Neural Networks, provides powerful models that are well capable to represent with high accuracy and high stability the thermochemical quantities stored in a 4D lookup table for hydrogen flames.
- However, for the 6D lookup table for dutch natural gas, the ANN model for the source term of the progress variable shows some non-physical peaks for stoichiometric-non diluted-adiabatic mixtures for small fluctuations in mixture fraction. Nonetheless, this is only present in a confined area in the table. Therefore, these instabilities should not cause important numerical issues when CDF simulations are being performed.
- The previous non-physical peaks are not a surprise in the sense that the ANN technique can be categorized as an optimization problem, where there is a target and the ANN tries to predict the target with the smallest error possible. Therefore, in order to improve these model, we must inform the model in advance with the physics that underlies in the lookup table generation.
- Although in this project paid a lot of attention on choosing carefully the activation function and rescaling adequately the data, alternative techniques must be employed in order to make these models physically-informed.
- For the other thermochemical quantities stored in the 6D lookup table, the ANNs are quite good in understanding the flamelet data in terms of accuracy and stability.
- The high accuracy and stability obtained employing ANNs are achieved at considerably lower storage requirement in comparison to the lookup tables studied in this report. Hence, they achieve the goal of reducing substantially the memory requirement of conventional lookup tables.

- Likewise, obtaining highly stable models depends on the choice of the activation functions in hidden and output layers. Additionally, using more than one hidden layer helps to achieve this goal.
- Finally, the FGM and its extensions (for instance, DA-FGM approach) rely on accurate generation of steady and unsteady flamelets. Creating inaccurate flamelets could lead to nonphysical results as it was found out in the middle of this project. Therefore, choosing carefully the extinction strain rate is one of the most important decisions in order to obtain good results when the FGM approach and its extensions are used.

Finally recommendations and future research for the current topic are provided as follows:

- With the new 4D lookup table for hydrogen flames, it may be a good idea to repeat the CFD simulation made using ANSYS Fluent in [20]. Better results and faster convergence may be expected.
- As it was mentioned in section 6.2.1, one way of informing the model about the physics that underlies on the table is to refine the table in the zone where the nonphysical peaks were found. Adding the values of the output at $\widetilde{\xi''^2} = 0.05$ may be enough to force the model to lower these sharp peaks and, at the same time, avoid increasing substancially the table size. This also would have a directly consequence in not increasing so much the training time of the ANNs.
- For the current project, the large amount of data contained in the lookup tables allows avoiding the use of regularization techniques such as dropout, batch normalization and others, which prevent the Machine Learning models to suffer overfitting. Nevertheless, for other cases where the data set is smaller than the ones considered in this project, such techniques must be used in order to avoid overfitting.
- As it was mentioned in section 3.3, the inputs and outputs dimensionality can be in principle as many as we want (there is no restriction). However, higher dimensionality on inputs and outputs implies a more complex model and a higher risk of overfitting. For large data sets as the ones considered in this project, this does not represent an issue. Nevertheless, for smaller data sets and high input dimensionality, overfitting will be more likely to occur. This is called *Curse of dimensionality*. In such cases, dimensionality reduction techniques such as Principal Component Analysis (PCA) should be applied.
- Finally, CFD simulations using the ANNs would be necessary in order to study whether they improve the convergence due to their smooth representations(in comparison to the linear interpolation) and, whether this improvement in convergence balances the expected overhead due to higher retrieval time using ANNs.

Appendix A

Other ANN architectures for the source term of the progress variable

The results for other ANN architectures used for the Source term of the progress variable are given as follows:

- Architecture 36-36-36-36:

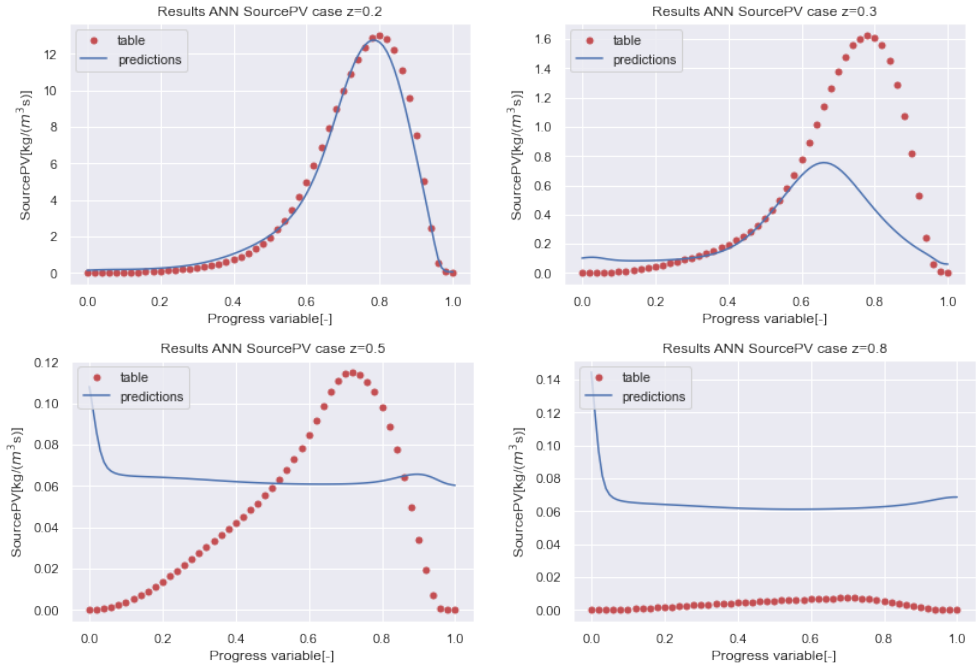


Figure A-1: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table. Architecture 36-36-36-36.

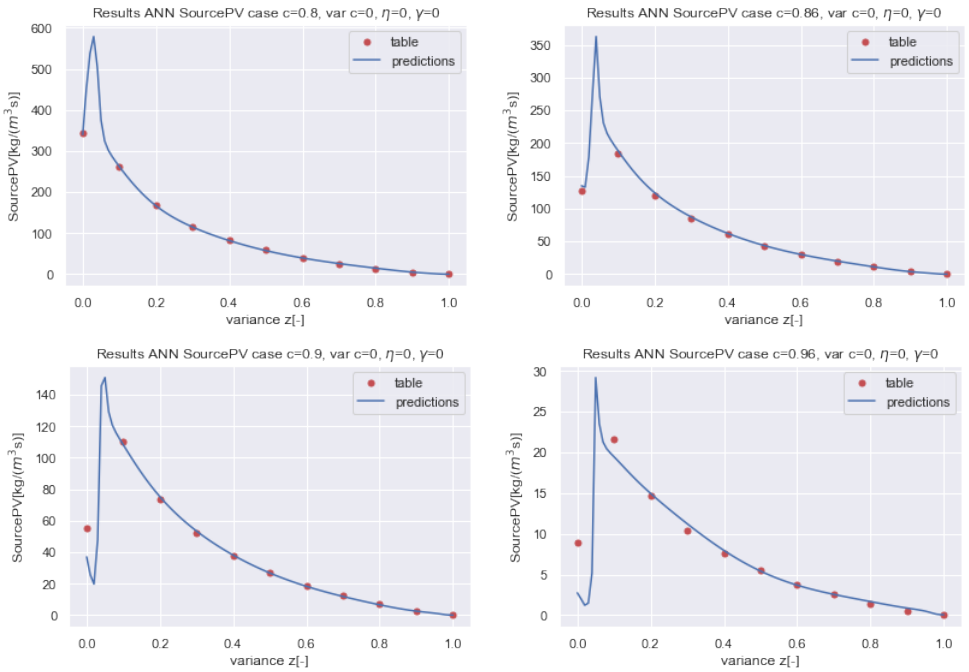


Figure A-2: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table. Architecture 36-36-36-36.

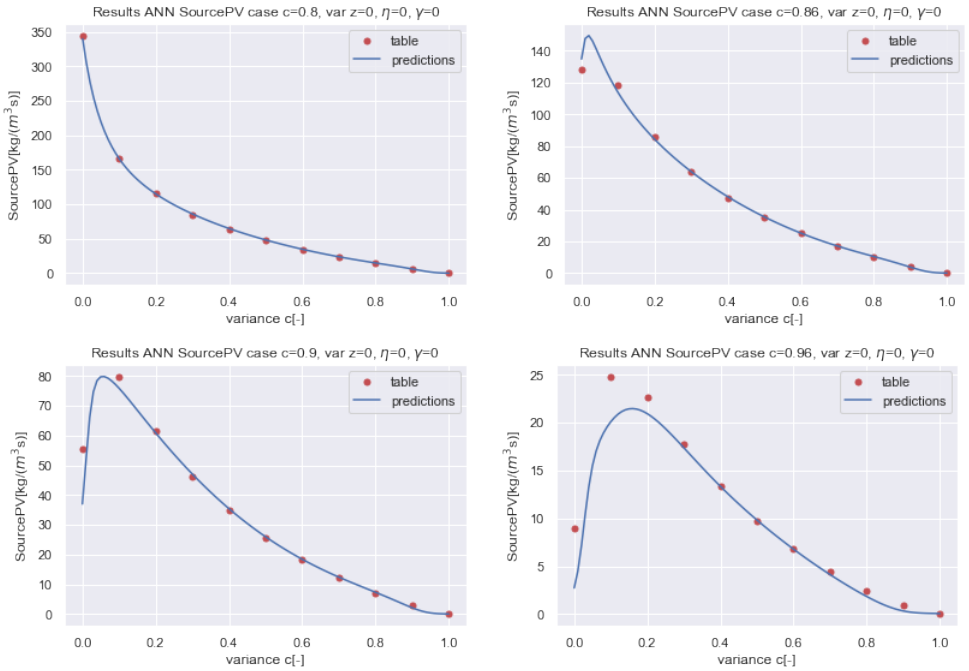


Figure A-3: Case 3: Source term of the progress variable with respect to the variance of scaled progress variable for different values of mean scaled progress variable. 6D Table.Architecture 36-36-36-36.

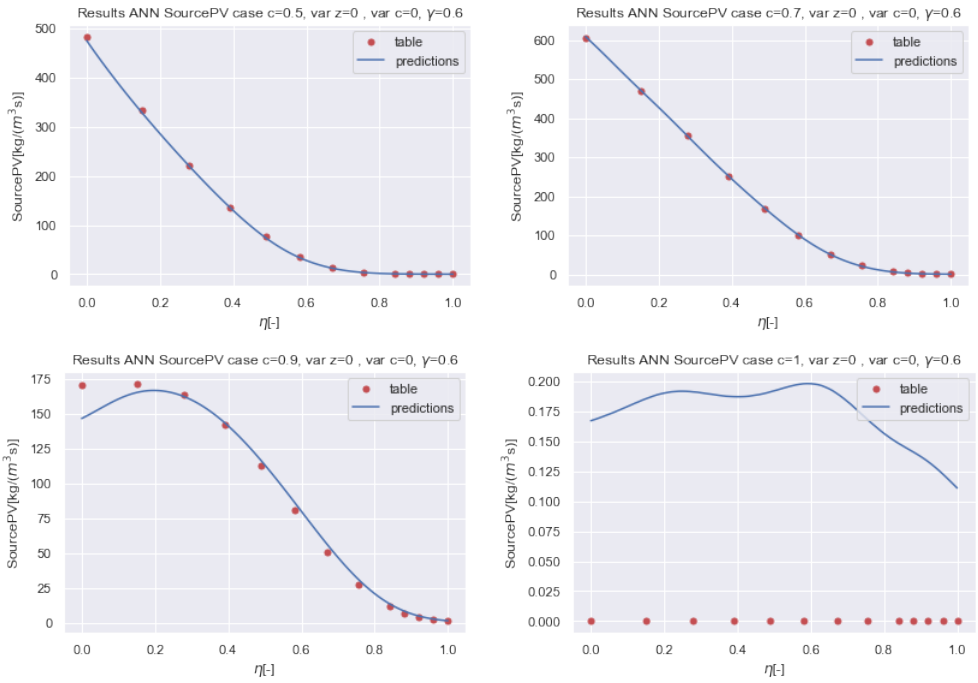


Figure A-4: Case 4: Source term of the progress variable with respect to enthalpy loss factor for different values of the scaled progress variable. 6D table. Architecture 36-36-36-36.

- Architecture 16-16-16-16-16:

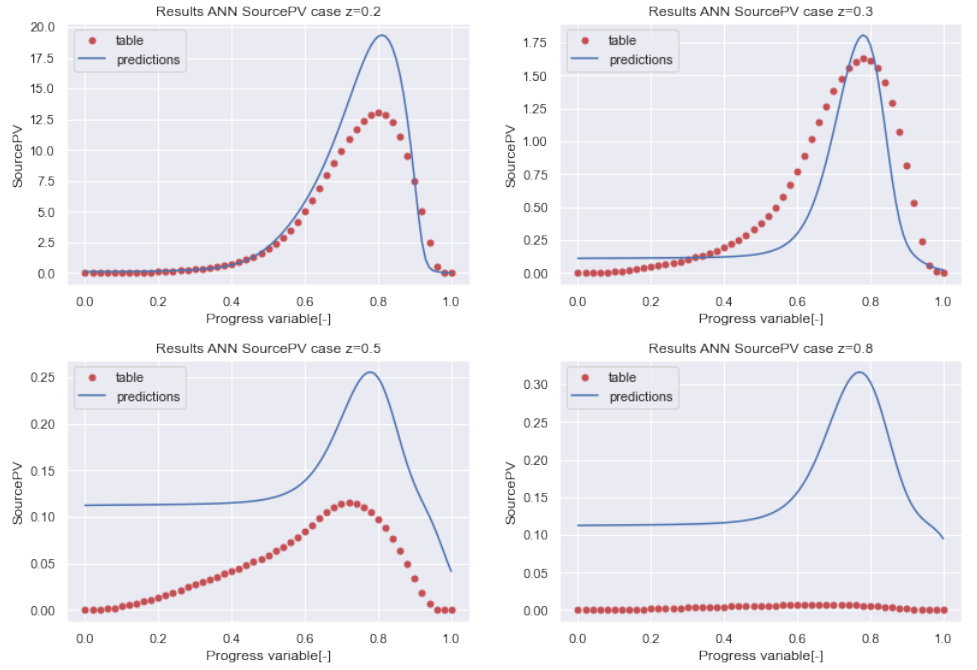


Figure A-5: Case 1: Source term of the progress variable with respect to mean scaled progress variable for different values of mean mixture fraction. 6D Table. Architecture 16-16-16-16-16.

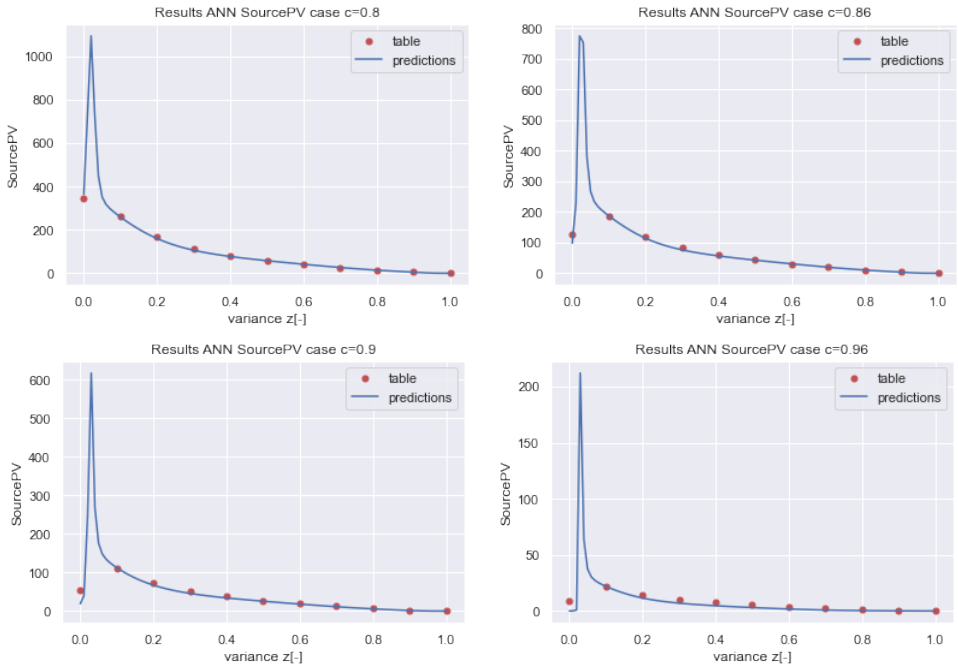


Figure A-6: Case 2: Source term of the progress variable with respect to the variance of mixture fraction for different values of mean scaled progress variable. 6D Table. Architecture 16-16-16-16-16.

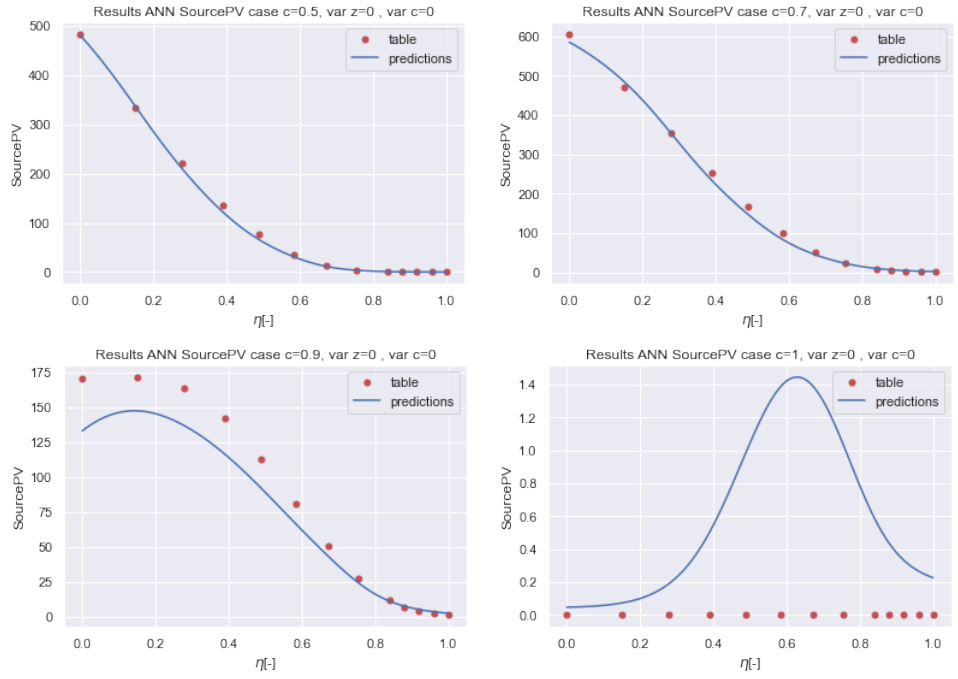


Figure A-7: Case 4: Source term of the progress variable with respect to enthalpy loss factor for different values of the mean scaled progress variable. 6D table. Architecture 16-16-16-16-16.

Bibliography

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. USA: Oxford University Press, Inc., 1995. Chap. 4. ISBN: 0198538642.

[2] J Blasco et al. "A self-organizing-map approach to chemistry representation in combustion applications". In: *Combustion Theory and Modelling* 4.1 (2000), pp. 61–76. DOI: 10.1088/1364-7830/4/1/304. eprint: https://doi.org/10.1088/1364-7830/4/1/304. URL: https://doi.org/10.1088/1364-7830/4/1/304.

[3] Mathis Bode et al. "Adaptive chemistry lookup tables for combustion simulations using optimal B-spline interpolants". In: *Combustion Theory and Modelling* 23 (Feb. 2019), pp. 1–26. DOI: 10.1080/13647830.2019.1583379.

[4] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019. ISBN: 9781999579517. URL: https://books.google.nl/books?id=0jbxwQEACAAJ.

[5] Alberto Cuoci. *Machine Learning for data-based predictive models in combustion with detailed kinetics*. ERCOFTAC Course. Dec. 2019.

[6] A. D'Ausilio, I. Stankovic, and B. Merci. "Numerical study on the importance of the turbulent inlet boundary condition and differential diffusion in a turbulent H2/N2/air jet diffusion flame". In: *Combustion Science and Technology* 191.1 (2019), pp. 109–125. DOI: 10.1080/00102202.2018.1452402. URL: https://doi.org/10.1080/00102202.2018.1452402.

[7] A. Ronald Gallant and Halbert White. "On learning the derivatives of an unknown mapping with multilayer feedforward networks". In: *Neural Networks* 5.1 (1992), pp. 129–138. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80011-5.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016. Chap. 6.

[9] Xu Huang. "Measurements and model development for flameless combustion in a lab-scale furnace". PhD thesis. Delft University of Technology, 2018. DOI: https://doi.org/10.4233/uuid:209bd573-be15-46be-9bda-310d0e21d481.

[10] Xu Huang et al. "Modelling of MILD combustion in a lab-scale furnace with an extended FGM model including turbulence-radiation interaction". In: *Combustion and Flame* ().

[11] Matthias Ihme, Alison Marsden, and Heinz Pitsch. "Generation of Optimal Artificial Neural Networks Using a Pattern Search Algorithm: Application to Approximation of Chemical Systems". In: *Neural computation* 20 (Mar. 2008), pp. 573–601. DOI: 10.1162/neco.2007.08-06-316.

[12] Matthias Ihme, Christoph Schmitt, and Heinz Pitsch. "Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame". In: *Proceedings of the Combustion Institute* 32 (Dec. 2009), pp. 1527–1535. DOI: 10.1016/j.proci.2008.06.100.

[13] Soteris A. Kalogirou. "Artificial intelligence for the modeling and control of combustion processes: a review". In: *Progress in Energy and Combustion Science* 29.6 (2003), pp. 515–566. ISSN: 0360-1285. DOI: https://doi.org/10.1016/S0360-1285(03)00058-3. URL: https://www.sciencedirect.com/science/article/pii/S0360128503000583.

[14] Juan Li et al. "An updated comprehensive kinetic model of hydrogen combustion". In: *International Journal of Chemical Kinetics* 36.10 (), pp. 566–575. DOI: https://doi.org/10.1002/kin.20026. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/kin.20026.

[15] Christopher M.Bishop. *Pattern recognition and Machine learning*. Springer, 2006. Chap. 5,6. ISBN: 978-0387-31073-2.

[16] U. Maas and S.B. Pope. "Implementation of simplified chemical kinetics based on intrinsic low-dimensional manifolds". In: *Symposium (International) on Combustion* 24.1 (1992). Twenty-Fourth Symposium on Combustion, pp. 103–112. ISSN: 0082-0784. DOI: https://doi.org/10.1016/S0082-0784(06)80017-2. URL: https://www.sciencedirect.com/science/article/pii/S0082078406800172.

[17] U. Maas and S.B. Pope. "Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space". In: *Combustion and Flame* 88.3 (1992), pp. 239–264. ISSN: 0010-2180. DOI: https://doi.org/10.1016/0010-2180(92)90034-M. URL: https://www.sciencedirect.com/science/article/pii/001021809290034M.

[18] G. Maragkos, P. Rauwoens, and B. Merci. "Assessment of a methodology to include differential diffusion in numerical simulations of a turbulent flame". In: *International Journal of Hydrogen Energy* 40 (2015), pp. 1212–1228.

[19] W. Meier et al. "Characterization of Turbulent hytVAir Jet Diffusion Flames by Single-Pulse Spontaneous Raman Scattering". In: *Combustion Science and Technology* 118.4-6 (1996), pp. 293–312. DOI: 10.1080/00102209608951983. URL: https://doi.org/10.1080/00102209608951983.

[20] M.M. Nelissen. "Modelling Differential Diffusion in Turbulent Non-Premixed Hydrogen Flames". MA thesis. Delft University of Technology, 2021. DOI: https://repository.tudelft.nl/islandora/object/uuid:45cd3604-4634-4b9d-a32d-3a7e79318dc4?collection=education.

[21] Jeroen van Oijen and Philip Goey. "Modelling of Premixed Laminar Flames using Flamelet-Generated Manifolds". In: *Combustion Science and Technology - COMBUST SCI TECHNOL* 161 (Dec. 2000), pp. 113–137. DOI: 10.1080/00102200008935814.

[22] C. Pantano. "Direct simulation of non-premixed flame extinction in a methane–air jet with reduced chemistry". In: *Journal of Fluid Mechanics* 514 (2004), pp. 231–270.

[23] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[24] Heinz Pitsch. *Machine-Learning Based Techniques for Error Estimation in Combustion Modelling*. ERCOFTAC Course. Dec. 2019.

[25] Thierry Poinsot and Denis Veynante. *Theoretical and Numerical Combustion*. 2012. Chap. 4. ISBN: 978-2-7466-3990-4.

[26] W.J.S. Ramaekers. “Development of flamelet generated manifolds for partially-premixed flame simulations”. English. PhD thesis. Mechanical Engineering, 2011. ISBN: 978-90-386-2750-2. DOI: 10.6100/IR716707.

[27] Rishikesh Ranade et al. “An Efficient Machine-Learning Approach for PDF Tabulation in Turbulent Combustion Closure”. In: *Combustion Science and Technology* 193.7 (2021), pp. 1258–1277. DOI: 10.1080/00102202.2019.1686702. URL: https://doi.org/10.1080/00102202.2019.1686702.

[28] Deep Ray and Jan Hesthaven. “An artificial neural network as a troubled-cell indicator”. In: *Journal of Computational Physics* 367 (Apr. 2018). DOI: 10.1016/j.jcp.2018.04.029.

[29] Mengmeng Ren, Shuzhong Wang, and Dirk Roekaerts. “Numerical study of the counterflow diffusion flames of methanol hydrothermal combustion: The real-fluid effects and flamelet analysis”. In: *The Journal of Supercritical Fluids* 152 (2019), p. 104552. ISSN: 0896-8446. DOI: https://doi.org/10.1016/j.supflu.2019.104552. URL: https://www.sciencedirect.com/science/article/pii/S089684461930021X.

[30] L.M.T. Somers. “The simulation of flat flames with detailed and reduced chemical models”. English. PhD thesis. Mechanical Engineering, 1994. DOI: 10.6100/IR420430.

[31] *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[32] Sergios Theodoridis. *Machine learning: A Bayesian and optimization perspective*. Academic Press, 2020. Chap. 1,18. ISBN: 9780128188033.

[33] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition*. Academic Press, 2009. Chap. 3.7,4,11. ISBN: 978-1597492720.

[34] J.A. van Oijen, F.A. Lammers, and L.P.H. de Goey. “Modeling of complex premixed burner systems by using flamelet-generated manifolds”. In: *Combustion and Flame* 127.3 (2001), pp. 2124–2134. ISSN: 0010-2180. DOI: https://doi.org/10.1016/S0010-2180(01)00316-9. URL: https://www.sciencedirect.com/science/article/pii/S0010218001003169.

[35] Yan Zhang et al. “Large eddy simulation of spray combustion using flamelet generated manifolds combined with artificial neural networks”. In: *Energy and AI* 2 (Nov. 2020), p. 100021. DOI: 10.1016/j.egyai.2020.100021.

Glossary

List of Acronyms

ANN	Artificial Neural Network
MP	Multilayer Perceptron
UAT	Universal approximation theorem
AI	Artificial Intelligence
SOM	Self-Organizing Map
FGM	Flamelet Generated Manifold
ML	Machine Learning
DL	Deep Learning Neural Network
PCA	Principal Component Analysis
DA-FGM	Diluted Air FGM
RANS	Reynolds Average Navier-Stokes
LES	Large Eddy Simulation
GD	Gradient Descent
SGD	Stochastic Gradient Descend
EBP	Error Back Propagation
SGD	Stochastic or Online Gradient Descent
NS	Navier-Stokes equations
DNS	Direct Numerical simulations
LES	Large Eddy Simulations
CFD	Computational Fluid Dynamics
CSV	Comma Separated Value
RELU	Rectified Linear
PDF	Probability Density Function
ILDM	Intrinsic Low Dimensional Manifold

MILD	Moderate or Intense Low Oxygen Dilution
MSE	Mean Squared Error
WSGG	Weighted-Sum-of-Grey-Gases
RTE	Radiative Transfer Equation
TRI	Turbulent Radiation Interaction

List of Symbols

μ	Learning Rate