# Convolutional Neural Networks for Heat Conduction

Sidharth Tadeparti Vishal V. R. Nandigana

September 7, 2021

**Abstract**

This paper presents a data-driven approach to solve heat conduction problems, in particular 2D heat conduction problems. The physical laws which govern such problems are modeled by partial differential equations. We examine temperature distributions of conductors that have square geometry subjected to various boundary conditions, both Dirichlet and Neumann. The data consists of images of these distributions in a semi-continuous form. Conventionally, such problems may be solved analytically or using numerical methods which can be computationally expensive. We attempt to use Image-Based Deep Learning algorithms such as encoder-decoders and variational auto-encoders which do not involve the physical laws of the problem. We also study the efficacy of deterministic models against probabilistic models and the feasibility of using image-based deep-learning methods for engineering applications.

**Keywords**: Heat Conduction, Deep Learning, Image-Based Algorithm, Convolutional Neural Networks.

## 1 Introduction

Heat Transfer Problems involve the study of the movement of heat between systems, it occurs through four modes advection, conduction, radiation, and convection. Advection is the movement of the phase itself, a transport phenomenon that depends on the motion of the medium. Conduction is heat transfer through vibration between particles in physical contact, while convection is a transport phenomenon where heat flows by virtue of a moving phase. Radiation is the transfer of heat through electromagnetic radiation. In this paper, we focus on conduction problems, in particular 2D steady-state heat conduction.

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{1}$$

$$0 \leq x \leq a,$$
$$0 \leq y \leq a.$$

The physics of steady-state heat conduction problems are governed by an elliptic partial differential equation(1), where $\Delta$ is the Laplacian operator and $\mathbf{u}$, the temperature. The physical state of the systems can be determined by solving the partial differential equation along with certain boundary conditions relevant to the problem. Analytical solutions of a steady-state heat conduction problem can be determined for simple geometries, however, for more complicated geometries, an analytical solution may not attainable. With the advent of the digital computer, numerical methods have been employed to solve partial differential equations, in such methods, the solution for the problem is computed at discrete points in space rather than a general solution for every point in space, in the process of discretization errors are introduced which are satisfactory provided number of discretized points is sufficiently large for a given problem.

The finite element analysis (FEA) is a numerical method that divides the domain of the solution into a number of smaller elements using a mesh, these elements may be of different dimensions depending on the nature of the problem. The partial differential equations which govern the physical situation are applied over each element in a weighted integral form, this ensures that the fundamental conservation laws are satisfied in the obtained numerical solution. The variables at the discrete points in space are obtained from a linear system formed through linearization of the weighted integrals. The solution obtained through this method has errors from discretization and linearization, both of which increase with increasing sizes of elements. As the complexity and the size of the problems increase, elements of smaller size are needed to achieve a numerical solution of sufficient accuracy. As a result, the number of elements increases causing an increase in the size of the linear system increasing the to compute time of the solution. Some physical situations may require a mesh of extremely fine resolution, for example, chaotic phenomenon such as turbulence, in such cases the number of variables to be solved for and the number of equations is large, such systems may take several days to be solved at a High-Performance Computing Facility which employs parallel computing, such simulations are beyond the capabilities of a personal computer. [1]

With improvements in computational infrastructure and growth in GPU computing, deep learning can be employed in a variety of applications, *engineering*, and in particular *thermal engineering* is no exception. Such approaches can have utility in problems where simulations are computationally expensive. Data from simulations can be used to train deep learning models. However, engineering applications are unique in the sense that there can be less data available compared to other domains. As demonstrated in [2] physics informed neural networks help overcome the limitation posed by the limited availability of

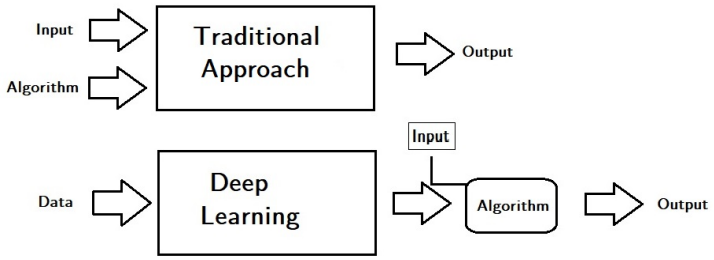



Figure 1: Traditional Approach, Deep Learning Approach

data. In this paper, we examine a non-physics informed, image-based approach. [3] Establishes that neural networks are universal function approximators that are capable of mapping any Borel measurable function between different dimensional spaces. This property of multi-layered neural networks finds utility in heat conduction problems where an analytical solution may not be available, yet an approximated solution for the temperature distribution can be obtained by training a neural network. In this paper, we attempt to *teach* a neural network the underlying rules and heuristics which accompany the problem without actually incorporating them into the algorithms. Encoder-Decoder architectures are primarily used in this paper, 3 channel images of temperature distributions, and the corresponding boundary conditions serve as the data. Digital images are essentially stored as numbers in an array, these images are often stored in an efficient manner using encoding techniques to reduce the size of the Image. In this paper, we use encoding - decoding algorithms that try to understand the physics of heat transfer implicitly. The reconstruction capabilities of encoder-decoder networks are demonstrated in [4],[5]. Traditionally such networks have found use in denoising and compression, for example in [6], where X-rays are denoised using denoising auto-encoders. In this paper, such networks are used to generate temperature distributions from incomplete temperature distributions which include the boundary conditions. The use of convolution neural networks to generate temperature distributions is demonstrated in [7], it focuses on predicting temperature distributions for various geometries with identical boundary conditions. However [7] does not establish whether CNN's are capable of learning the fundamental laws of heat transfer and whether CNN's are capable of generating temperature distributions for arbitrary boundary conditions. In this paper, we examine the generative capabilities of image-based algorithms(CNN's) both deterministic and probabilistic for different boundary conditions. We further evaluate the feasibility of image based algorithms against non-image based deep learning algorithms (Figure 1).

**Problem Setup & Description:** The problem we attempt to solve in this paper is the 2D conduction problem, 2D conduction is governed by equation(1).

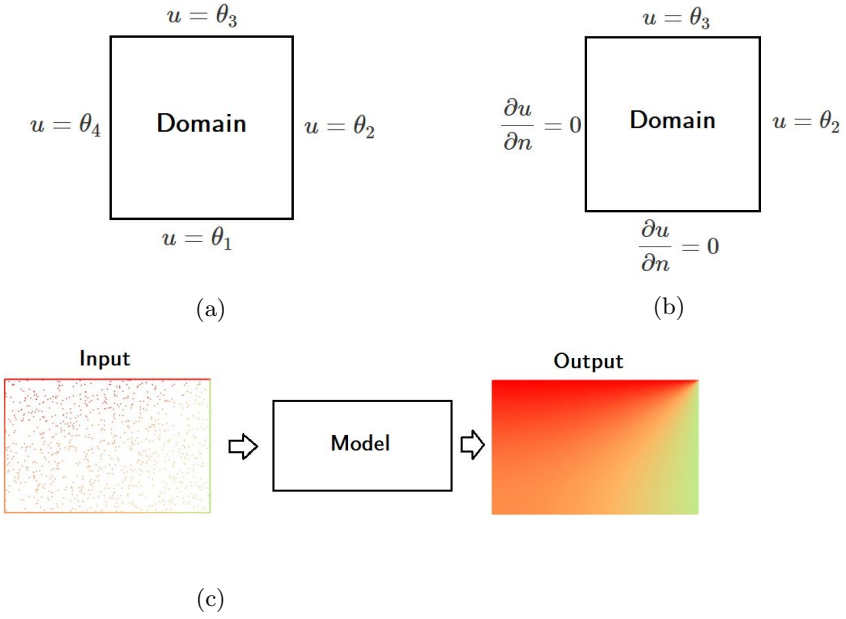



Figure 2: Boundary Conditions and Black Box Model

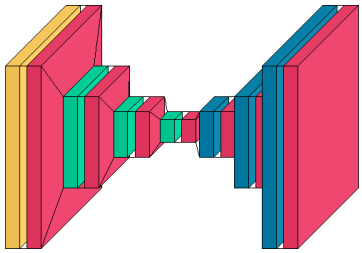
To obtain a unique solution to this problem, boundary conditions have to specified. This problem is solved for two specific cases, Dirichlet boundary conditions Figure(2a) and Neumann boundary conditions Figure(2b).

We attempt to create a deep learning model that takes inputs in the form of boundary conditions and incomplete temperature distributions and complete the temperature distributions as shown in Figure(2c). The data set consists of 20 pairs of input and output images which have been generated using data from commercial simulation software.

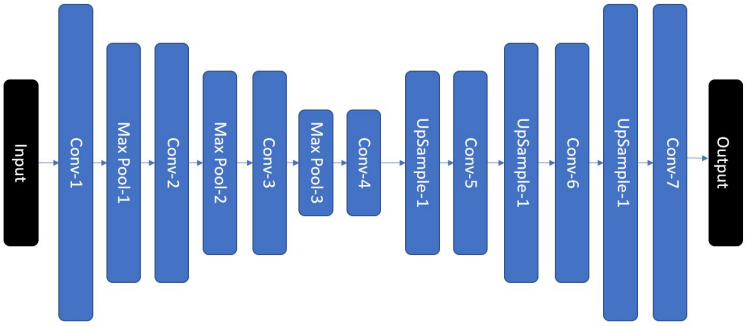
## 2 Deep Learning Models

Dimensionality reduction techniques are commonly used in applied statistics and machine learning to remove redundant features that add no additional information. In such techniques data is transformed from a higher dimensional to a lower-dimensional space having independent features(Figure 3a), sometimes referred to as the latent space. Such Transformations may be linear or non-linear in nature.

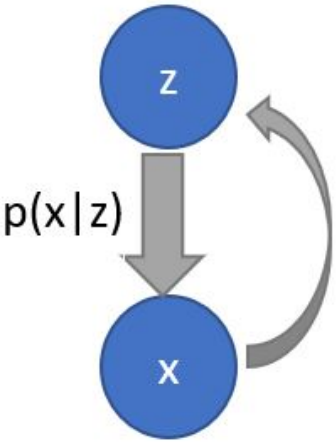
The simplest such dimensionality reduction technique is principal component analysis (PCA), where there is a linear transformation to a latent space where features are independent [8]. A major drawback of PCA is its linear nature, something with auto-encoders with non-linear activation functions seem



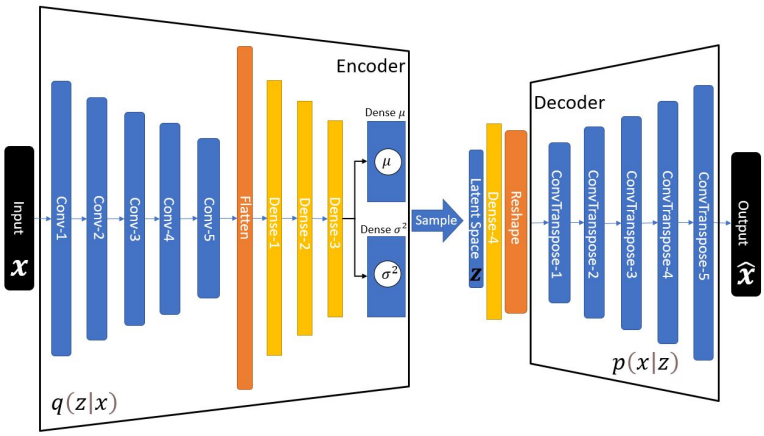
(a) Encoder Decoder Model




(b) Autoencoder




(c) Variational Autoencoder




(d) Variational Autoencoder Architecture

Figure 3: AutoEncoder DL Models

to overcome. This non-linearity can be exploited along with the universal function approximate ability of neural networks to learn and understand patterns seen in heat transfer. Two variants of encoder-decoder models are studied, a probabilistic and a deterministic, the probabilistic variant is called a variational auto-encoder(VAE), has its latent space variables to be stochastic as opposed to the traditional auto-encoder model which has single-valued latent space variables. In addition to the Image-based algorithms, a non-image based approach is also applied. A Distributed Artificial neural network (DANN) is trained using data in a numerical form rather than an image form [9]. The primary objective of training this model is to understand the efficacy of image-based deep learning approaches against non-image based approaches.

## 2.1 Autoencoder

This model uses CNN's which have non-linear activation functions at each layer. The encoder portion of the model reduces the dimensionality of the input data, in the process captures crucial information about the temperature distribution while discarding redundant information. From the lower-dimensional space (latent space) the decoder generates images using up-sampling. Auto-encoders function on the principle that data concentrates and clusters around low-dimensional manifolds, which is represented in the latent space [10]. Auto-encoders which have noise in the input are called denoising auto-encoders, the auto-encoder in this case. This is demonstrated by the reconstruction capability of denoising auto-encoders studied in [11]. The architecture of the model used is depicted in Figure(3b), it is an under complete All convolution layers used, have same padding, which ensures the out has the same width and height as the input. The max pooling layers reduce the width and height of the input to half. The input to the model is a three-channel image which is of size (256 X 256 X 3) and the output is an image of the same size. The reconstructed output is further processed using a normalization scheme to obtain a temperature distribution.

The loss function used in this model is the *mean square loss*:

$$MSE_i = \frac{1}{W \cdot H \cdot D} \sum_{j=1}^{W} \sum_{k=1}^{H} \sum_{l=1}^{D} |\hat{x_i}(j,k,l) - x_i(j,k,l)|^2$$

Where $MSE_i$ is the mean square loss associated with image $i$, ($W$ X $H$) is the dimension of the image, D is the number of channels, $\hat{x}$ is the reconstructed output, while $x$ is the input. $j$ and $k$ represent the location of pixels and $l$ represents the channel and $x_i(j,k,l)$ is the value of a particular pixel. The loss does not possess a regularization term. This is due to fact that the auto-encoder being used is under complete and the size of the bottleneck is small, making it resilient to over-fitting, this is particularly important in engineering applications where the availability of data is limited. The adadelta optimization algorithm is used. The activation function used to create non -linearity in the model is ReLU.

## 2.2 Variational Autoencoder

A variational auto-encoder(VAE) as the name suggests is a probabilistic version of the traditional auto-encoder, it is introduced by [12]. The latent space of a VAE consists of random variables that have a priori, in the model used in this paper the priori is a normal distribution. A normal distribution can be parameterized by a mean and variance. Using Convolutional neural networks and artificial neural networks the probabilistic model can be represented in a deterministic framework by sampling from a normal distribution whose parameters are the output of the encoder. From the sampled latent space a decoder generates the output which on further processing gives a temperature distribution.

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{2}$$

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(z-\mu)^2/2\sigma^2} \tag{3}$$

$$L = E_{q(z|x)} \log p\left(x|z\right) - KL\left(q\left(z|x\right)||p\left(z\right)\right) \tag{4}$$

In Figure(3c) z is the latent space that we wish to find, x is the data that corresponds to the latent space. We wish to find $p(z|x)$, However computing $p(z|x)(2)$ is not possible as the integral to compute $p(x)$ might be intractable. Hence variational inference is used to estimate $p(z|x)$ using a distribution $q(z|x)$. This is done by assuming the latent space variables to be Gaussian, serving as a priori for the random variable z(3). Figure(3d) highlights the architecture of the model used. Much like the auto-encoder, the VAE captures crucial information about the input($x$) using CNN's. The output from the CNN's are flattened and passed through an ANN which terminates with the mean and variance of a parameterized normal distribution. Value for the latent space is obtained through a sampling of values from the obtained normal distribution which then passed through a series of transpose convolutional layers which generate the output $\hat{x}$). Upon further processing, the output temperature distribution is obtained. Similarity between two probability distributions is given by the KullbackLeibler divergence or the relative entropy denoted by $KL(P||Q)$, where $P$ and $Q$ are two probability distributions. The loss function (equation(4)) consists of two components, a reconstruction component and KL Divergence component, where $E_{q(z|x)}$ is the expectancy with respect to $q\left(z|x\right)$. While p(z) is given by equation(3). The optimization function used is Adam(adaptive moment estimation), and the activation function used is ReLU. Once again a regularization term is not used as the auto-encoder is under complete and care is taken to limit the size of the bottle-neck to avoid memorization of features.

## 2.3 Distributed Artificial neural network

In this model given presented, [9], each spatial point of each piece of input data is trained using a single DANN, the activation function used is ReLU. The information doesn't transmit within each input as the weights are calculated for each point independent of neighboring points. The minimization objective for the algorithm is the mean-squared loss. The model is mathematically illustrated below.

$$DANN = \forall \left\{ \begin{array}{ll} 0 & \text{if } x \leq 0 \\ \oint_{\Omega} \int_{j=1}^{m} \left(h_{j_i} + b_{2_i}\right) djd\Omega_i & \text{if } x > 0 \end{array} \right\}$$

$$h_j = \forall \left(W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i}\right)$$

$$MSE_i = \frac{1}{m} \sum_{n=1}^{m} \left|T_{\text{output }_i} - T_{Act_i}\right|^2$$

Where $i$ is a spatial coordinate, $x$ is the input data at each coordinate, $h$ is the hidden cell state, $W_{1_i}, b_{1_i}$ and $W_{2_i}, b_{1_i}$ are the weight and bias matrices for hidden-hidden and input-hidden connections. $\oint_{\Omega}$ is the integral over the engineering geometry of interest, $j$ is the training sample and $m$ is the total number of training samples. Further, $x_i$, is called features. The boundary condition for each grid point $i$, for sample $j$, is denoted as $b_{2_i}$ Figure(4).

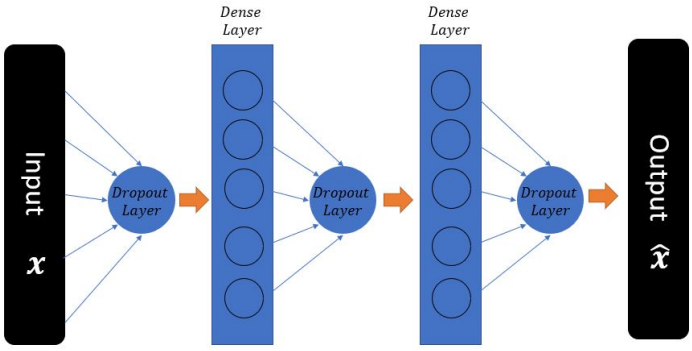



Figure 4: DANN Schematic

# 3 Results

The data used in this paper is in the form of images. In thermal engineering applications, the amount of data available is limited, hence only 20 temperature distributions are used to train the deep learning models. The temperature distributions are generated using commercial simulation software for both Neumann and Dirichlet boundary conditions. The obtained distributions are verified using a symbolic computational tool. While the algorithms are trained with reconstruction in mind, i.e. the targets and the inputs are the same completed temperature distribution, while during testing the incomplete temperature distribution(Figure 5) is used to generate completed temperature distributions. This is justified by the fact that denoising-encoders have the potential to recognize and reconstruct corrupted versions of data from a learned distribution. The incomplete consists of the boundary conditions and temperature at a few randomly sampled points, this is done to aid the deep-learning model to learn the relevant features and to ensure a large portion of the network doesn't become inactive due to pixels that have null values. Furthermore, the final temperature distribution output is obtained by the use of a normalization scheme to generate colorized images(Figure 5). This is done by using luminance as a metric to detect patterns. Luminance is given by equation(5) where R, G, B refer to the values of the respective channel at a pixel on the image.

$$L_i = 0.21 \cdot R + 0.72 \cdot G + 0.07 \cdot B \tag{5}$$

## 3.1 Simulation Data Set Generation

A square domain of side length $2m$ is subjected to boundary conditions. Figure (6b) shows the verification of results obtained using simulation software. The numerical solution is now found for 20 such cases for both Dirichlet and Neumann type boundary conditions. The obtained solution is then converted to

(a) Input During Testing (b) Output after Colourizing

Figure 5: Comparison between input during testing and output after contour colourizing

image form programmatically and used as input. Only 20 data points are used to reflect the limited availability of data in a thermal engineering domain. 15 of these form the training data-set and 5 of these form the testing data-set.

## 3.2 Comparison Between CNN and PPRNN

The performance metric used for the CNN models is SSIM(structural similarity index measure). SSIM has been a standard metric to test the quality of images against a benchmark quality. SSIM varies from 0 to 1, where 0 implies structural dissimilarity and 1 implies perfect similarity. SSIM measures structural similarity between two black and white images .SSIM is given by equation(6) where x and y are two sliding windows of the same size, $\mu$ represents the average, $\sigma$ represents the variance/covariance and ($C_1$ ,$C_2$) are stabilising constants to ensure bounded values when the denominator is weak.

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{6}$$

SSIM is used for the auto-encoder and variational auto-encoder models as the generated temperature distribution is formed using a normalization scheme, hence the final output is representative of the patterns rather than the exact temperature value at a point. Moreover, the temperature distribution(image) will be identical for two boundary conditions which are in proportion given the nature of the laplace equation.Since SSIM is a metric for black and white images, the obtained processed output is converted to black and white. Hence the SSIM serve as a measure of the how well the image-based algorithm reconstructs patterns from the input and extracts features pertaining to heat transfer.

| Metric | Autoencoder | Variational Autoencoder |
|---|---|---|
| SSIM | 0.7797 | 0.6671 |

Table 1: Results for the Dirichlet Boundary Condition

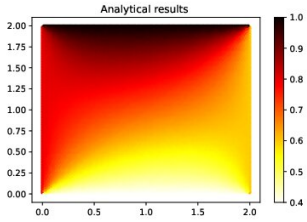
## 3.3 Heat Conduction Problem and DL model image solution comparison

### 3.3.1 Autoencoder

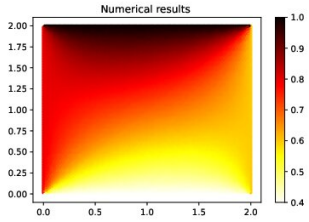
Under a Dirichlet boundary condition the output temperature distribution is as shown in Figure(6). The processed output indicates primitive feature extraction with the target distribution's shape not being perfectly reflected. The average SSIM for the reconstruction is **0.7797**. The training MSE is 0.0185 and validation MSE is 0.0200.The model is trained for 1000 epochs, convergence is seen in Figure(6) . The model takes 1s/epoch to train and 85ms/sample for a prediction.

### 3.3.2 Variational Autoencoder

Figure(7) shows the processes output after 2000 epochs of training, while the loss has converged Figure(7), the final training loss is relatively high.This is reflected in the fact that the output has a nearly constant output.The trained model has a reconstruction loss of 41824.4668 and a kl-loss of 12.3372. An SSIM of **0.6671** is obtained. The training time per epoch is 8s and it takes around 300 ms for a prediction.

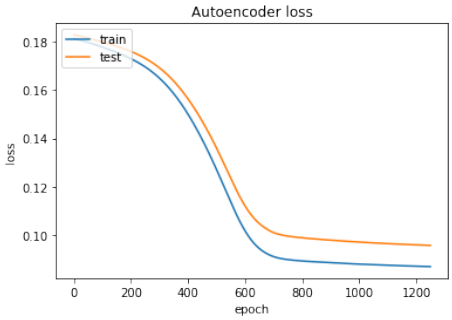



(a) Analytical Solution

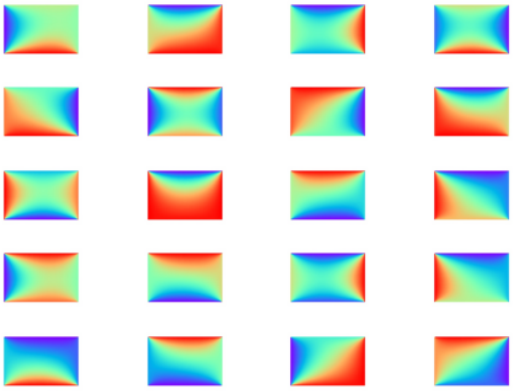



(b) Numerical Solution

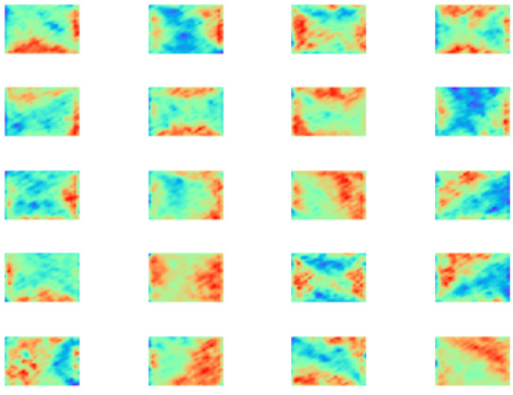
Comparison between Analytical and Numerical Solution of sample heat conduction problem




(c)

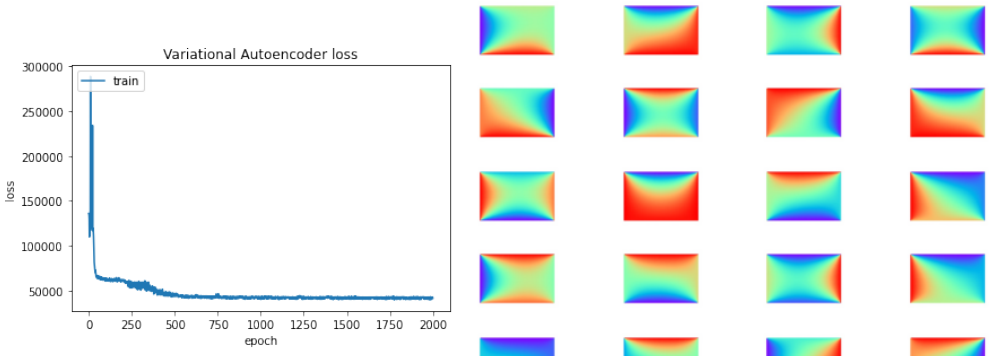


(d)



(e)

Figure 6: Comparison of losses




(a)

(b)

(c)

Figure 7: Comparison of losses

### 3.3.3 DANN

The DANN algorithm as presented in [9] shows maximum errors of less than **1%** for both Dirichlet and Neumann boundary conditions under reasonable physical situations. Hence generating the temperature distributions almost perfectly. This establishes the superior capabilities of the DANN over the image-based models. When the DANN is supplied with data pertaining to a significantly smaller domain, of side length 2mm, where temperature gradients are much larger, the DANN output provides results which may be closer to reality than a numerical solution to the laplacian for a set of boundary conditions. The DANN algorithm is trained for 200 epochs and a loss of $10^{-6}$ is obtained. Figure(8) depicts the results. The obtained output in Figure(8) is further analysed. While the numerical or the analytical solution to a partial differential equation may present a mathematical interpretation, the ground truth may be very different due to other physical effects. In the case the temperature gradients are large, it can lead to other effects,apart from conduction. While image based models learn features directly from the results of the partial differential equations, the DANN may provide more realistic output. Under Dirichlet boundary conditions in which the temperature gradients are high and asymmetrical, the material may not maintain its integrity, mixing and ignition may take place. In the DANN output of Figure(8), the center has values which are intermediate i.e. in between the highest and lowest boundary condition values, while the outer portions have relatively larger temperatures. Also due to disintegration of the medium, convective effects may also be seen in addition to conductive effects courtesy of high temperature gradients. This results in a circular pattern spread over a larger region than initially, similar to the 2D cross-section flame. This inference may be subjected to experimental verification for such asymmetrical boundary conditions.

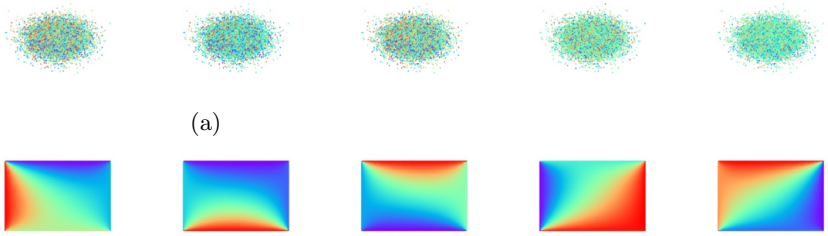
## 3.4 DL model image solution comparison for Neumann Boundary Condition

### 3.4.1 Autoencoder

The output temperature distribution under a Neumann boundary condition is given by Figure(9).The model seems to remotely capture the patterns associated with the target distribution. The average SSIM for the reconstruction is **0.7786**. The training MSE is 0.0058 and the validation MSE is 0.0068. Training is stopped at 1000 epoch, convergence is seen in Figure(9). The model training time is 1s/epoch and the prediction time is 89ms/sample.

### 3.4.2 Variational Autoencoder

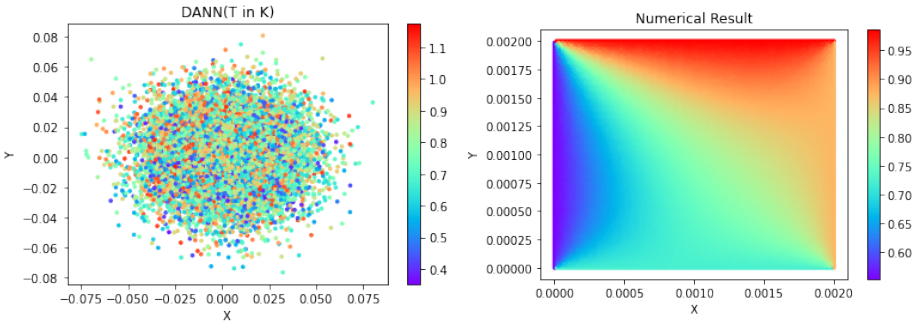
The processed output after 2000 epochs is given by Figure(9) . Similar to the Dirichlet case, the VAE model gives more or less a constant output . The SSIM is **0.7535**, which is largely due to similarity in shapes of black and white images of the target and the output. The reconstruction loss is 42838.4883 and the



(a)

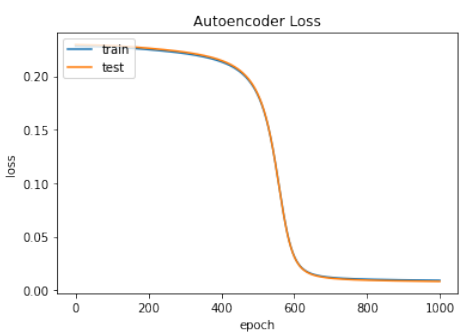


(b)

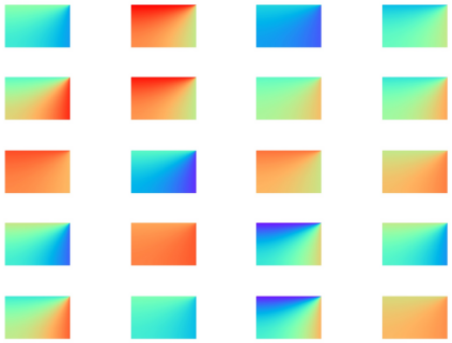



(c) (d)

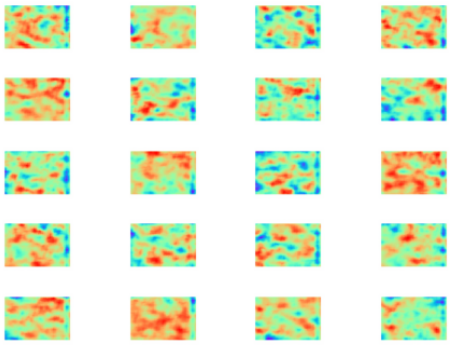
Figure 8: Comparison of losses



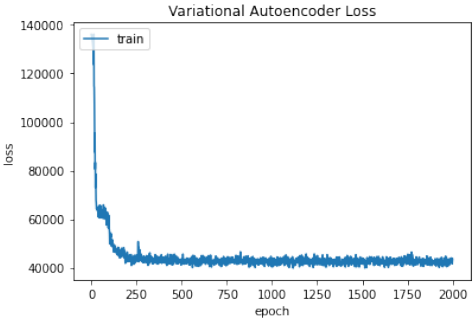

(a)



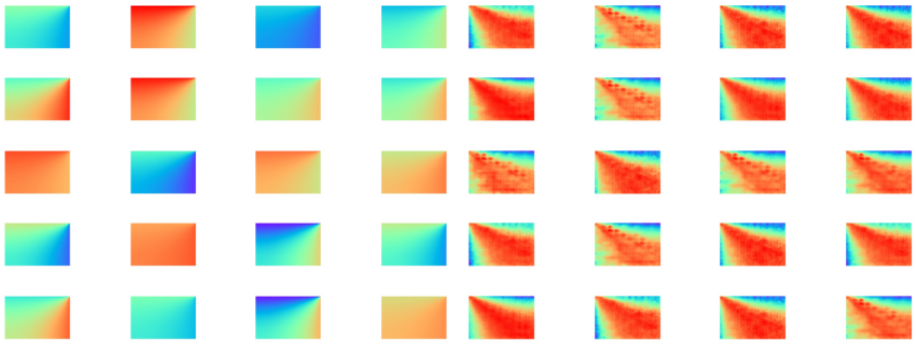
(b)



(c)




(d)



(e)

(f)

Figure 9: Comparison of losses for Neumann Boundary conditions

| Metric | Autoencoder | Variational Autoencoder |
|---|---|---|
| SSIM | 0.7786 | 0.7535 |

Table 2: Results for the Neumann Boundary Condition

kl-loss is 13.1811. The model takes 11s per epoch to train and 200 ms to make a prediction.

# 4 Summary

In this paper the efficacy of convolutional neural networks for a data-driven approach to deep learning has been studied. Images are generated using numerical solutions of PDE's and this is reflected in learning process of the image based models. The performance of the image based models is quantified using SSIM. The SSIMs obtained for the auto-encoder and the variational auto-encoder do not indicate satisfactory reconstruction. A simple examination of the outputs for the image based methods indicates that the models are unable to extract features accurately with the given data, especially in the case of the variation-auto-encoder. With better quality data-sets with a larger number of training samples greater feature extraction can be achieved, however the efficacy of such encoder-decoder networks for generative purposes remain limited, hence its capacity to learn complex features may be restricted. Generative adverserial networks may serve as better generative technique for image based modelling as demonstrated in several applications from other domains. As seen Image based modelling relies on the solution from partial differential equations, hence even in cases where length scales are small, the model will predict a similar output as when length scales are normal, discounting for effects other than conduction. The discrepancy is seen in the results pertaining to the DANN, which shows a significantly different output to the target are small length scales, the results obtained indicate that the non-image based provides a more realistic representation which image-based algorithms may not be able to provide. A finer interpretation of these results is a matter of further experimental investigation.

# 5 References

**List**

[1] Klaus-Jrgen Bathe. Finite element procedures. Klaus-Jurgen Bathe, 2006.

[2] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017.

[3] Kurt Hornik, Maxwell Stinchcombe, Halbert White, et al. Multilayer feedforward net-works are universal approximators. Neural networks, 2(5):359366, 1989.

[4] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 29, pages 28022810. Curran Associates, Inc., 2016.

[5] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Man-zagol, and Lon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12), 2010.

[6] Lovedeep Gondara. Medical image denoising using convolutional denoising autoen-coders. In 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), pages 241246. IEEE, 2016.

[7] Jiang-Zhou Peng, Xianglei Liu, Nadine Aubry, Zhihua Chen, and Wei-Tao Wu. Data-driven modeling of geometry-adaptive steady heat transfer based on convolutional neural networks: Heat conduction, 2020.

[8] Jonathon Shlens. A tutorial on principal component analysis. arXiv:1404.1100, 2014. arXiv preprint

[9] Dasari Ananyananda, Deepak Somasundaram, and Vishal Nandigana. Deep learning for engineering problems. Bulletin of the American Physical Society, 65, 2020.

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

[11] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data generating distribution, 2014.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.