inzva DLSG

Introduction to Deep Learning

Week 1

Contents

1 Introduction

As you read these lines, we assume you are already familiar with some fundamental notions of mathematics. Additionally, since you have the opportunity to participate in this study group, it's assumed that you have basic knowledge of machine learning algorithms and concepts.

The prerequisite knowledge for this course can be described as familiarity with the concepts in university-level courses on linear algebra, calculus, and probability theory. You can refer to this book for detailed coverage of these subjects. A basic understanding of these concepts will help you grasp most of the topics we will cover. You don't need to master every detail, but spending time with these concepts will enhance your understanding over time.

These notes are not a textbook chapter nor equivalent to university course notes. No wonder these notes tend to help one fully understand certain concepts, but the aim should be to gain intuition about the concepts. I do not think there is a unique way that everyone follows and gains expertise in this field. Since artificial intelligence is a 'very' multidisciplinary field, it's subfields are also contain a lot of different theory within. One might not prefer to dive into the theories and can take practical part in their focus. The choice of the way is up to you. However, along the way, we can dive into some mathematical concepts and if you get lost, you should not be worry about it. If one does not get lost, they will never have a chance to discover the path. So, embrace the journey of discovery as you navigate through the landscape of deep learning. May the force be with you.

1.1 A Journey Through Time: Neural Networks from the 1950s to Today

The story of neural networks is a fascinating one, marked by periods of both great excitement. Here, we will have a glimpse into this history:

1.1.1 Early Days (1940s-1960s):

In 1943, the concept of neural networks took a big leap forward thanks to the work of Warren McCulloch and Walter Pitts. Their groundbreaking paper[17] introduced a simple model of a neural network. This model used electrical circuits as an analogy for how neurons in the brain might function. Although this early model was basic, it laid the foundation for what would become a powerful tool in artificial intelligence - the artificial neural network. Their work essentially started the conversation about how machines could learn and process information in a way inspired by the human brain.

In 1949, a scientist named Donald Hebb made a crucial contribution to the understanding of neural networks with his theory of Hebbian learning. This theory proposes a mechanism for how learning happens in the brain. Here's the key idea: "Neurons That Fire Together, Wire Together". Hebb theorized that when two neurons fire (activate) at the same time, the connection between them strengthens. This strengthening is believed to be caused by changes in the synapses, the junctions[1] where neurons communicate.

This concept is fundamental to how neural networks learn. By simulating this process of strengthening connections between neurons that are repeatedly activated together, artificial neural networks can learn to identify patterns and improve their performance over time. Hebb's theory provided a crucial piece of the puzzle for building machines that could learn and adapt in a similar way to the human brain.

In the 1950s, the world of neural networks took a step from theory to practice. Pioneering researchers at IBM embarked on a significant endeavor: creating the first computer simulations of neural networks. These early attempts were like building the first airplanes - immature but crucial for future advancements.

The 1950s and 1960s were a time of both promise and frustration for neural networks. Simple models called **Perceptrons** emerged, offering a glimpse of the potential for artificial intelligence inspired by the brain. However, this early optimism was fired back by the work of Marvin Minsky and Papert in the 1960s. They discovered that Perceptrons had a fundamental limitation: they could only learn to classify data that could be separated by a straight line. This meant they couldn't handle more complex patterns in real-world data. This realization exposed a significant obstacle for the field, leading to a period of decline in research funding and interest in neural networks. Even though Perceptrons showed promise, their limitations became a roadblock, forcing the field to take a step back and wait for new approaches and breakthroughs to emerge.

1.1.2 The AI Winter (1960s-1980s):

The 1960s to the 1980s were a chilly time for neural networks, often called the "AI Winter." Early models like Perceptrons seemed promising, but they couldn't handle complex problems. Back then, computers weren't strong enough to train more complex networks. Since there wasn't much progress and less money for research, interest in neural networks slowed way down. This wasn't the end of the story though, because new discoveries and improvements were just around the corner.

[1] a place or point of meeting

1.1.3 The Renaissance (1980s-1990s):

The 1980s and 1990s witnessed a revival for neural networks, often termed the "Renaissance." A key turning point came in 1986 with the rediscovery and widespread use of backpropagation by Rumelhart, Hinton, and Williams. This technique allowed researchers to train more complex networks with multiple layers, leading to a surge in interest and advancements. Additionally, the 1980s saw the birth of **Convolutional Neural Networks** (inspired by the structure of the visual cortex in the brain). CNNs were specifically designed for image recognition tasks, further expanding the capabilities of neural networks. This period marked a significant shift from the limitations of the past, paving the way for even greater breakthroughs in the years to come.

1.1.4 The Neural Networks Strikes Back(1990s-Present)

Developments in late the 1990s was beginning of a forever transforming era for the field of neural networks. This period was fueled by two key factors. First, the rise of powerful **GPUs** significantly accelerated the training process for complex models, allowing researchers to explore previously unimaginable architectures. Second, the success of **AlexNet**[16] in the 2012 **ImageNet**[26] competition sparked a revolution. This win showcased the great potential of deep learning, leading to breakthroughs across various domains. Fueled by the advancements in architectures, neural networks began achieving **state-of-the-art performance** in tasks like **computer vision**, **natural language processing**, and **speech recognition**, paving the way for a wave of innovative applications that continue to shape our world today.

1.2 A Brief Introduction to Neural Networks

When we use a **deep neural network** to analyze data or extract information from data, it is called **deep learning**. Deep neural networks are a type of machine learning models. These models are currently the most **powerful** and **versatile** machine learning tools available. As a result, deep networks have become dominant in many areas of artificial intelligence and computer science, such as computer vision, natural language processing, and speech recognition. They're also widely used in fields like **healthcare**, **science**, **economics**, **mathematics**, and **medicine**. In fact, it's difficult to find a field that doesn't benefit from deep learning in some way.

1.2.1 Data Types

Deep learning is a **data-driven approach**, meaning we need data to use it effectively. Think about the word "data." What comes to mind first? An Excel spreadsheet? A collection of images? Perhaps even a book? The truth is, almost anything can be considered data for a specific task. We can even create our own unique forms of data, known as modalities, and use them to train our models. While the possibilities for data are vast, there are some important considerations... Here, we'll explore some common and intriguing data types encountered in deep learning, moving beyond traditional classifications.

Structured vs. Unstructured Data

Structured data is organized in a predefined format, like tabular data found in spreadsheets or databases. Rows represent data points, and columns represent features or attributes. This structured nature makes it easy for processing and analysis using traditional methods.

Unstructured data lacks a predefined format and often contains rich information. Examples include:

- **Text:** Text data, such as books, articles, and social media posts, is inherently unstructured. Techniques like **Natural Language Processing** are used to extract meaningful information from text data, enabling applications like **sentiment analysis**, **translation**, and **summarization**.
- **Images:** Images are unstructured data represented as pixel values in a grid format. Deep learning models, such as **Convolutional Neural Networks**, are highly effective in extracting features and patterns from images for tasks like **object detection**, **image classification**, and **image segmentation**.
- **Audio:** Audio data, such as speech or music, is another form of unstructured data. Deep learning techniques, including **Recurrent Neural Networks** and newer architectures like **Transformers**, are used to process audio signals for applications like **speech recognition**, **music generation**.

End-to-End Learning with Deep Learning

One of the key advantages of deep learning is its ability to handle unstructured data directly through **end-to-end learning**, which reduces the need for extensive pre-processing. Traditional machine learning often requires manual feature extraction and engineering, but deep learning models automatically learn relevant features from raw data.

For instance, in image recognition, CNNs automatically detect edges, textures, and shapes without manual intervention. Similarly, in NLP, models like Transformers capture complex relationships between words and phrases directly from raw text. This end-to-end learning approach not only simplifies the workflow but also often leads to better performance by leveraging the richness of unstructured data.

Understanding the differences between structured and unstructured data and utilizing deep learning's capabilities to process unstructured data effectively is crucial for advancing applications in various fields. By leveraging deep learning, we can unlock valuable insights and build powerful models that can learn from and adapt to the complexities of real-world data.

1.2.2 Model Architectures

In this course we will learn different types of **neural network architectures**. Here are the general architectures you might encounter. We will just learn some of them.

- **Multilayer Perceptron (MLP)**: The simplest form of neural network architecture where information moves in one direction, from input nodes through hidden nodes to output nodes. That is why it is sometimes called **feedforward neural networks**.

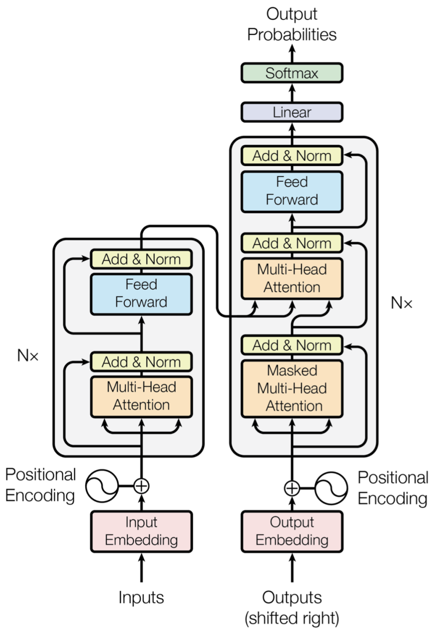

Figure 1: Transformers architecture[31]

- **Convolutional Neural Networks (CNN)**: They are usually used for image processing tasks. However, they can roughly process every kind of signal if you set it accordingly. They basically learn local spatial hierarchies of features through convolutional layers.

- **Recurrent Neural Networks (RNN)[25]**: These networks are designed to recognize patterns in sequential data that have long-range dependencies.(e.g. time series, sound waves, text). Long Short-Term Memory (LSTM)[12] and Gated Recurrent Unit (GRU)[1] are popular variants of RNNs used to address the **vanishing gradient** problem.

- **Transformers[31]**: They are originally introduced for natural language processing tasks, transformers rely on self-attention [31] mechanisms to weigh the importance of different words in a sentence. They have revolutionized language modeling and have been adapted to various sequential data tasks beyond NLP. After CNNs dominated the image domain for a long time, Vision Transformer (ViT)[3] model was introduced. We have seen that this model can outperform CNNs in many image-related tasks, employing a similar approach to transformers used in the NLP domain.

- **Graph Neural Networks(GNNs)[27]:** Graph Neural Networks (GNNs) change how we understand and study complex relationships in data. They treat data as connected graphs, which helps in tasks like understanding social networks or predicting molecular structures. GNNs learn by looking at how things are connected and use special layers to pass information efficiently between parts of the graph. They can see both small and big patterns in the data, making them useful for many different jobs like learning representations and making predictions in various fields.

- **Neural Differential Equations (NDEs)[2] :** Neural Differential Equations represent a novel approach to modeling dynamic systems using neural networks.

Instead of discrete layers, NDEs define continuous transformations based on differential equations, allowing for adaptive computation and efficient modeling of continuous-time dynamics.[13]

- **State Space Models:** State space models[7] (SSMs) have a long history in control theory, where they represent dynamic systems through hidden variables called "states." These states describe the dynamics of the system. In deep learning, SSMs are combined with deep neural networks. The deep network learns the hidden states, allowing the model to capture complex relationships in sequential data. Recent advancements in SSMs have shown them to be a very powerful method for modeling long-range dependencies. Examples of well-known SSM-based models in the literature include S4[6], S5[29], and Mamba[5].

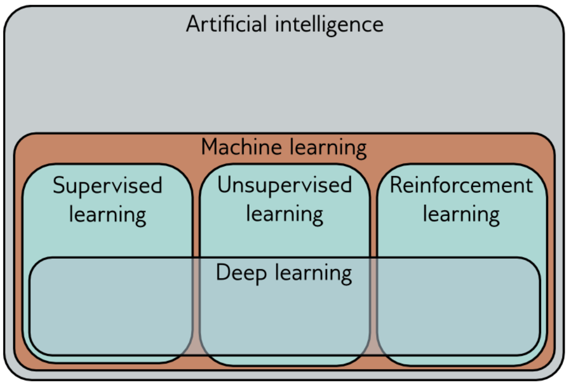

Figure 2: Deep learning finds applications across various subfields of machine learning such as supervised learning, unsupervised learning, and reinforcement learning.[18]

1.2.3 Common Tasks

Deep neural networks can be applied to common tasks like regression and classification. They are particularly powerful for handling complex or unstructured data. We will cover the most of the following tasks in this course through the lens of neural networks:

- **Image Classification:** One of the most fundamental tasks in computer vision, image classification involves categorizing images into predefined classes or categories. CNNs are commonly used for this task, with model named AlexNet[16], VGG[28], and ResNet[9] are the first models that gave hope to all artificial intelligence community.

- **Object Detection:** Object detection involves identifying and locating objects within images or videos. Models like Faster R-CNN[21], YOLO (You Only Look Once)[20] are used for object detection tasks.

- **Semantic Segmentation:** In semantic segmentation, the goal is to assign a class label to each pixel in an image, segmenting the image into different regions corresponding to different objects or classes. U-Net[24] is one of the most popular architectures for semantic segmentation.

- **Natural Language Processing (NLP):** NLP is not a specific task but umbrella term for the tasks of understanding human language. Tasks in NLP include sentiment analysis, named entity recognition, text classification, machine translation, text generation, and question answering, etc. Models like RNNs, Transformers, and their variants are commonly used for NLP tasks.

- **Speech Recognition:** Speech recognition involves converting spoken language into text. RNNs and CNNs are commonly used for speech recognition tasks. Some models used in along with models like WaveNet[30] and DeepSpeech[8].

- **Generative Modeling:** Generative modeling tasks involve generating new data samples from a given dataset. Variational autoencoders (VAEs), generative adversarial networks (GANs), Normalizing Flows, autoregressive models like PixelCNN, PixelRNN and recently popular diffusion models like DDPM[11], Latent Diffusion Models[23] are commonly used for generative modeling tasks.

Throughout the course, we will encounter many of these models and tasks. As we delve into the tasks, explore various architectures, talk about improving training and some best practices, you will train your own models for related tasks, ultimately gain self-confidence and take the first step for becoming a deep learning practitioner.

1.2.4 Some Famous Model Designs

There are incredibly clever designs for processing data, extracting information, and accomplishing specific tasks. Or, to put it another way, these designs solve particular problems in different ways. Here are some famous examples.

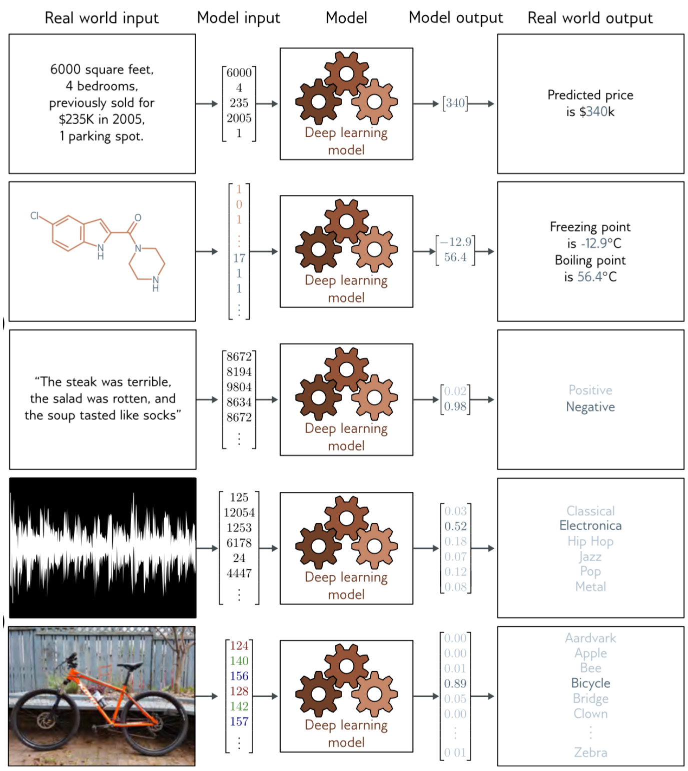

Figure 3: Some regression and classification tasks solved by deep learning models [18]

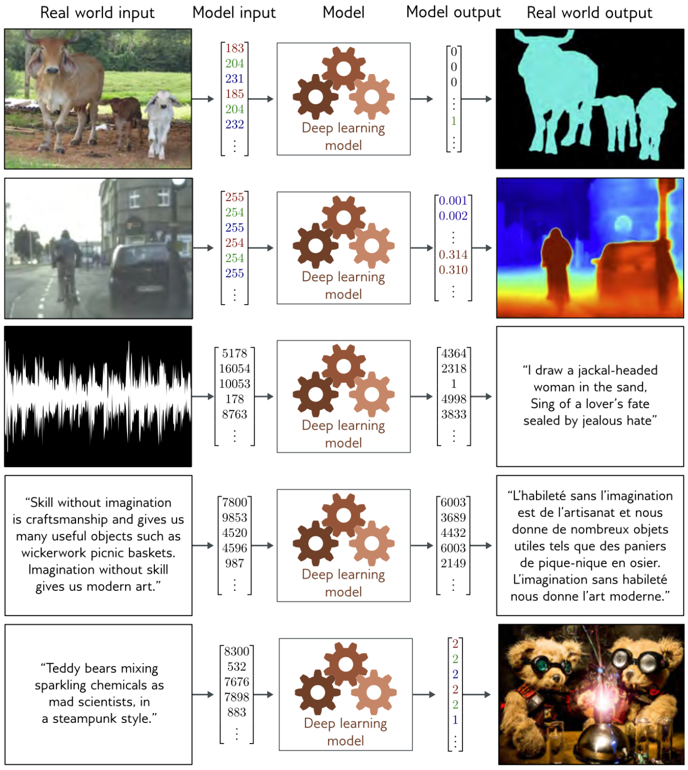

Figure 4: Image segmentation, monocular depth estimation, speech to transcript, machine translation, text-to-image generation, respectively[18]

If you're new to deep learning, the terminology might seem overwhelming at first. Don't worry! This is just to spark your interest and curiosity.

- **Residual Neural Network (ResNet):** ResNet is a deep convolutional neural network architecture that introduced the concept of **residual learning**. It addresses the problem of **vanishing gradients** in very deep neural networks by introducing shortcut connections, also known as **skip connections**, which allow gradients to flow more directly during training. These skip connections enable the network to learn residual mappings instead of directly fitting the desired underlying mapping. ResNet architectures have been widely adopted and have achieved **state-of-the-art** performance in various computer vision tasks, including image classification, object detection, and semantic segmentation.[10]

- **Generative Adversarial Networks (GANs):** GANs are a class of generative models consisting of **two neural networks**, a generator and a discriminator, trained simultaneously in a **minimax game** setting. The generator learns to generate data samples resembling real data, while the discriminator learns to distinguish between real and generated samples. GANs have found applications in image generation, style transfer, and data augmentation.[4][19]

- **Variational Autoencoders (VAEs):** Variational Autoencoders are generative models that learn to encode and decode data in a **latent space**. They consist of an **encoder network** that maps input data to a probability distribution in latent space and a **decoder network** that reconstructs the input data from samples drawn from this distribution. VAEs are trained using **variational inference** and are widely used for tasks such as data generation, dimensionality reduction, and unsupervised learning.[15][14]

- **Diffusion Models:** Diffusion models are **probabilistic generative models** that estimate the **likelihood of data** by simulating a **diffusion process**. They capture the dynamics of how information spreads through a system over time, allowing for the generation of high-quality samples and efficient inference.[11][23]

- **Normalizing Flows:** Normalizing Flows are a class of generative models that learn to transform a simple probability distribution, such as a Gaussian distribution, into a more complex distribution representing the data. These transformations are designed to be **invertible**, allowing for **efficient sampling** and **exact likelihood computation**. Normalizing Flows have gained popularity for their flexibility and ability to model complex data distributions.[22]

Let us talk about learning. What do we mean by learning? How do machines learn? Or do they really learn? How do we define learning? How do we learn? For decades, these questions have been asked by many scientists from various fields of science. All of them have their theories and intuitions about learning. We will be certainly interested in computer science perspective for a while. However, in the realm of artificial intelligence, various fields complement each other and leverage each other's insights to develop new ideas.

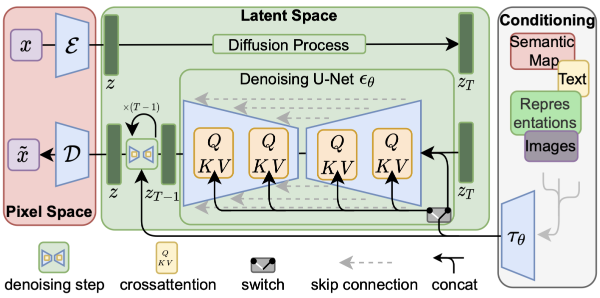

Figure 5: Latent diffusion models [23]

We, as humans, mostly learn by experience, study, or instruction. We observe patterns in our surroundings, connect different pieces of information, and enhance our understanding through **trial** and **error**. Similarly, in deep learning models, this learning process occurs by processing vast amounts of data to identify patterns and extract meaningful representations. These models iteratively **adjust their parameters** to **minimize** the difference between their predictions and the actual outcomes, particularly in the supervised case. Techniques such as **backpropagation** and **gradient descent** enable deep learning models to update their parameters, enhancing their performance on specific tasks. Soon, we will delve into the inner workings of deep learning models.

References

[1] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, 2016.

[2] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations, 2019.

[3] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[4] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks, 2014.

[5] Gu, A., and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

[6] Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces, 2022.

[7] Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., and Ré, C. Combining recurrent, convolutional, and continuous-time models with linear state-space layers, 2021.

[8] Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., and Ng, A. Y. Deep speech: Scaling up end-to-end speech recognition, 2014.

[9] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

[10] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

[11] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

[12] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[13] Kidger, P. On neural differential equations, 2022.

[14] Kingma, D. P., and Welling, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning 12*, 4 (2019), 307–392.

[15] Kingma, D. P., and Welling, M. Auto-encoding variational bayes, 2022.

[16] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012), F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, Curran Associates, Inc.

[17] Mcculloch, W., and Pitts, W. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics 5* (1943), 127–147.

[18] Prince, S. J. *Understanding Deep Learning.* MIT Press, 2023.

[19] Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[20] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection, 2016.

[21] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[22] Rezende, D. J., and Mohamed, S. Variational inference with normalizing flows, 2016.

[23] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.

[24] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation, 2015.

[25] Rumelhart, D. E., and McClelland, J. L. Learning internal representations by error propagation, 1987.

[26] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252.

[27] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks 20*, 1 (2009), 61–80.

[28] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

[29] Smith, J. T. H., Warrington, A., and Linderman, S. W. Simplified state space layers for sequence modeling, 2023.

[30] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio, 2016.

[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023.