Predictive Coding and Biologically Plausible Neural Networks

Bachelor Project

Anders Bredgaard Thuesen
s183926@student.dtu.dk

January 2022

Abstract

Contents

0.1 Introduction 4
0.1.1 Related work 4
0.2 Dataset 4
0.3 Feed Forward Neural Networks 4
0.4 Backpropagation 4
0.5 Biological constraints 5
0.6 LIF Neuron model 5
0.7 Predictive Coding 6
0.8 Variational Inference 6
0.9 Energy Based Models 6
0.10 Conclusion 6

0.1 Introduction

In the recent years, deep learning has shown impressive results due to the availability of massive parallel compute and huge amounts of data. From the biological inspiration of the neuron to the perceptron where data inputs are weighted, summed together and thresholded, several new modern architectures, like recurrent, residual and transformer neural networks have pushed the limits and achieved state of the art results in speech recognition, computer vision and natural language understanding. Despite of these networks being originally inspired by the brain, the backpropagation (backprop) algorithm for learning the weights and deep learning in general has been criticized for being biologically implausible [1]. This project will primarily be dealing with on of them: The weight transport problem which arises from the way backprop uses the connection weights in both the forward pass (inference) and the backwards parse (calculating the gradients), requiring that both forward and backward connections have symmetric weights and that information is able to flow backwards through the weights.

Besides having both philosophical as well scientific interest, studying the computational aspects of how the human brain processes sensory input might lead to great improvements in deep learning and artificial intelligence.

0.1.1 Related work

Write some stuff here about predictive coding, spiking neural networks as such.

0.2 Dataset

Before we can define our neural network and derive our backpropagation algorithm, we first need to define the dataset on which the network is trained and evaluated. For this we introduce a set of N pairs of datapoints, $\mathbf{x}_i \in \mathbb{R}^k$ and $\mathbf{y}_i \in \mathbb{R}^l$ where N is the size of our dataset, $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}$ for $i = 1 \dots N$. We notice, that both $\mathbf{x}_i$ and $\mathbf{y}_i$ can be vectors of possibly differently dimensions k and l respectively. Here $\mathbf{x}_i$ might represent eg. an image with its horizontal and channel axes flattened into a single one-dimensional vector, as we will encounter later when training on images of handwritten digits in the MNIST dataset [4]. In the case of supervised learning, our objective is from $\mathbf{x}_i$ to predict the corresponding label, $\mathbf{y}_i$ which amounts to a single scalar number from 0 to 9 in the MNIST dataset.

0.3 Feed Forward Neural Networks

We define the weight matrices of our feed forward neural network with $L - 1$ hidden layers as $\mathbf{W}^{(l)} \in \mathbb{R}^{m \times n}$ for $l = 0 \dots L$. We initialize the weights using He initialization, where the entries of the matrix $W_{ij}^{(l)}$ is draw from a normal distribution, $\mathcal{N}(0, 2/n)$ [2].

0.4 Backpropagation

The working horse of allmost all modern deep learning models is the back-propagation (aka. backprop) algorithm first popularized for training neural networks by Rumelhart, Hinton & Williams in 1986 [5]. The algorithms solves what is referred to as the *credit-assignment* *problem.* When learning the parameters of an artificial neural network we would like to know how changing a weight in the network contributes to the total loss, in order to change it in the direction that minimizes the loss. One naive way to do this would be simply to adjust a single random weight slightly, evaluate the new neural network on the dataset and observe the effect on the model loss. If the change leads to a decrease in loss, keep the change, otherwise repeat from the beginning. This would however be very computationally expensive, since the network would have to be evaluated on the entire dataset for each weight in the network. Fortunately, the backprop algorithm achieves this much more efficiently.

$$Z = XW^T \tag{1}$$

$$A = \sigma(Z) \tag{2}$$

0.5 Biological constraints

0.6 LIF Neuron model

Derivation of spiking rate as function of input current.

$$V(t) = (V(0) - j)e^{-t/t_{RC}} + j$$

$$1 = -je^{-t/t_{RC}} + j$$

$$1 - j = -je^{-t/t_{RC}}$$

$$\frac{1-j}{j} = -e^{-t/t_{RC}}$$

$$\frac{1}{j} - 1 = -e^{-t/t_{RC}}$$

$$1 - \frac{1}{j} = e^{-t/t_{RC}}$$

$$\log(1 - \frac{1}{j}) = \frac{-t}{t_{RC}}$$

$$t = -t_{RC}\log(1 - \frac{1}{j})$$

$$r = \frac{1}{-t_{RC}\log(1 - \frac{1}{j})}$$

[3]

0.7 Predictive Coding

0.8 Variational Inference

0.9 Energy Based Models

0.10 Conclusion

Bibliography

[1] Yoshua Bengio, Dong-Hyun Lee, Jörg Bornschein, and Zhouhan Lin. Towards biologically plausible deep learning. *CoRR*, abs/1502.04156, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. 2015.

[3] Eric Hunsberger and Chris Eliasmith. Spiking deep networks with lif neurons. 2015.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[5] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.