

# Deep Learning Final Project: Binary Neural Networks

Brandon Duderstadt

April 29, 2018

## 1 Motivation and Problem Statement

Deep neural networks are currently the top performing algorithm on a wide range of artificial intelligence tasks, ranging from object recognition, to complex system control, to natural language processing. Many of these tasks are critical components of real time robotic systems, and thus, deep neural networks have become increasingly popular in the robotics community. However, these networks come with a major drawback; their computational cost at inference time is prohibitively high. This renders the networks very slow on many embedded robotics applications, and unfit for use in scenarios requiring low resource, real time computation.

The main computation taking place in these networks is the multiply-accumulate operation, as each artificial neuron essentially computes a weighted sum of its inputs. Naturally, a reduction in the time cost of this operation would result in a huge savings in total network inference complexity. In particular, if the weights of each artificial neuron were constrained to either -1 or 1, the 32-bit floating point multiply-accumulate operations taking place at each neuron could be replaced with a much faster 1-bit XNOR-count operation.

Unfortunately, the optimization problem associated with solving for these weights is a 0-1 integer program, which is NP-Complete. This renders direct optimization methods ineffective. Furthermore, traditional nonconvex optimization methods, such as gradient descent, are difficult to apply here, since the XNOR operation does not have a continuous derivative. The resulting question is thus:

**How can one train a binary neural network such that they can take advantage of the speed increase related with using XNOR-accumulate operations as opposed to multiply-accumulate operations at inference time?**

## 2 Prior Work

Courbariaux et. al (2016) [1] present the current state of the art algorithm for learning these weights through a series of train-time approximations and weight-clipping. I will use this algorithm as a basis for my investigations.

## 3 Datasets

For my investigations, I will be using the MNIST [2] and Fashion-MNIST [3] datasets. I have selected these datasets since they are two of the canonical benchmark datasets for deep learning tasks.

## 4 Methods

The main module that I will be using in my investigations will be a **Binary Linear Unit** (BLU). The algorithm below defines the behavior of this unit during the forward pass:

---

**Algorithm 1** Binary Linear Unit Forward Pass

---

**Require:**  $X$  is the input tensor

**Require:**  $W$  is the non-binary layer weight matrix

**Require:**  $B$  is the non-binary layer bias matrix

**function** FORWARD( $X, W, B$ )

$W_b \leftarrow \text{Sign}(W)$

$B_b \leftarrow \text{Sign}(B)$

$X \leftarrow xW_b + B_b$

$X \leftarrow \text{Sign}(X)$

**return**  $X$

**end function**

---

The main idea to note here is that, during training time, the network stores its weights as 32-bit floats. However, during the training time forward pass, the network only **uses** the signs of the weights. In this way, the inference step of the BLU requires only binary weights.

One drawback of the above function is that it makes heavy use of the sign operation, which has derivative 0 almost everywhere. Thus, the backward pass of the BLU ignores the presence of the sign function, and passes the gradient straight through to the next operation in the operation tree. While this does not approximate the true gradient of the function, it allows the signal from backpropagation to continue through the network without dying out at the sign operations.

## 5 Experiments

## References

- [1] Matthieu Courbariaux and Yoshua Bengio. Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830, 2016.
- [2] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.
- [3] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.