**Machine Learning in Healthcare**

**HW 4 – Neural Network Models**

**Ofek Aloush  
307841742**

**Asaf Tzur**

**\_\_\_\_\_\_\_**

**Part I – fully connected layers**

**Task 2**

ReLU stands for rectified linear unit. It is an activation function and defined as the positive part of its argument:

Leaky ReLU, on the other hand is defined as:

Leaky ReLU has two main benefits:

1. First, it does not have parts with zero slope. ReLU has slope of zero in the negative part which causes dead neurons, meaning no learning is performed. In contrary to ReLU, Leaky ReLU always have at list a little slope which allows the gradient to flow. If for some reason the input of a neuron becomes bellow zero, in ReLU it can die, but in LReLU it will always has a bit of gradient left, so it can shift back to life over time.
2. Secondly, it is more balanced and therefore may learn faster.

More benefits?

**Task 4**

In Batch Gradient Descent (BGD) we take the all data for each step of gradient descent, but this is obviously problematic when data is too large. On the other hand, Stochastic Gradient Descent (SGD) takes one example of the data for each step. Since SGD use one example at a time, we can't implement vectorized implementation on it, which may slow down the computations.

Mini-Batch Gradient Descend (MBGD) is a mixture of BGD and SGD. We define a fixed number of training examples and use it as small batches.   
Using this method, we enjoy the both worlds: we don't use all the data at each step, but since we use several example we can use the vectorized implementation.

Check for more advantages/disadvantages.

**Part II – Convolutional Neural Network**

**Task 1 – 2D CNN**

Layers and filters: Make sure all of these are consider as layers.

|  |  |  |
| --- | --- | --- |
| Layer no. | Layer | Number of filters |
| 1 | Permute | 0 |
| 2 | Conv 2D | 64 |
| 3 | Dropout | 0 |
| 4 | Batch Normalization | 0 |
| 5 | Max Pooling 2D | 0 |
| 6 | Conv 2D | 128 |
| 7 | Dropout | 0 |
| 8 | Batch Normalization | 0 |
| 9 | Conv 2D | 128 |
| 10 | Dropout | 0 |
| 11 | Batch Normalization | 0 |
| 12 | Conv 2D | 256 |
| 13 | Dropout | 0 |
| 14 | Batch Normalization | 0 |
| 15 | Conv 2D | 256 |
| 16 | Dropout | 0 |
| 17 | Batch Normalization | 0 |
| 18 | Max Pooling 2D | 0 |
| 19 | Flatten | 0 |
| 20 | Dense | 0 |
| 21 | Dropout | 0 |
| 22 | Dense | 0 |
| 23 | Dense | 0 |

**Task 2 – Number of filters**

Reducing the number of filters increased performance. Instead of 29.7% accuracy, we got 37.1%.