**Report and analysis** – 30 points

o Note that the focus is on understanding and you have to provide meaningful/insightful analysis for what you expect from your experiments before doing them and explain what you have observed in your experiments for each of the cases for each of the networks as explained in the tasks.

Task 1

In your report, you need to provide explanations of your design choices and describe your neural networks clearly.

**Neural Network Design**

[**https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw**](https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw)

**https://towardsdatascience.com/exploring-activation-functions-for-neural-networks-73498da59b02**

For our neural network we’ve made several design choices that have led to performance exceeding 90% accuracy for our networks.

When choosing the size of layers to accurately model the problem we use empirically-derived rules-of-thumb. In particular, it’s well known that in order to reduce the chance of overfitting and chose a neural network of adequate size to model the problem at hand, the number of neurons in the hidden layer should ideally with the range of the size of neurons in the input layer and output layer. In this case [10, 256].

Activation functions allow us to model non-linear properties for modeling non-linear problems. The choice of activation functions is important because during the learning the backpropagation algorithm calculates the gradient of the activation function, so we can pass the maximum amount of the error though the network during back-propagation if we use ReLU which always has a derivative value of 1 or 0. over Sigmoid and Tanh because the max derivative of Sigmoid is 0.25, Tanh on the other hand has a max derivative of 1. This implies that during our backpropagation algorithm that ReLU consistently learns faster. In an unconstrained problem, ReLU in every layer proves to be empirically superior for learning. In this problem we explore the option of considering Sigmoid and Tanh in the hidden layers.

For the final layer SoftMax is chosen because we’d like our classifier to calculate a probability distribution used to calculate the loss and that determines the error that our backpropagation will exploit to correct the weights. The goal is that as the certainty of a classification increases, the corresponding class probabilities should decrease.

***Fully Connected Neural Network***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer # | Type | Neurons | Kernel Size | Activation | Input Shape |
| 1 | Input | 256 |  |  | 256 |
| 2 | Fully Connected | 128 |  | Relu |  |
| 3 | Fully Connected | 128 |  | Sigmoid |  |
| 4 | Fully Connected | 128 |  | Tanh |  |
| 5 | Output | 10 |  | Softmax |  |

***Locally Connected Neural Network (No Weights Shared)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer # | Type | Neurons | Kernel Size | Activation | Input Shape |
| 1 | Input | 256 |  |  | 16 x 16 x 1 |
| 2 | Convolutional 2D | 32 | 3 x 3 | Relu |  |
| 3 | Convolutional 2D | 64 | 3 x 3 | Tanh |  |
| 4 | Fully Connected | 128 |  | Sigmoid |  |
| 5 | Output | 10 |  | Softmax |  |

***Locally Connected Neural Network (Convolutional Neural Network)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Layer # | Type | Neurons | Kernel Size | Activation | Input Shape |
| 1 | Input | 256 |  |  | 16 x 16 x 1 |
| 2 | Convolutional 2D | 32 | 3 x 3 | Relu |  |
| 3 | Convolutional 2D | 64 | 3 x 3 | Tanh |  |
| 4 | Fully Connected | 128 |  | Sigmoid |  |
| 5 | Output | 10 |  | Softmax |  |

Task 2

Task 3