**Review Sheet for Test II (Monday Dec. 19th, 11:30 – 12:30)**

1. **Terminologies**

* Activation functions: logistic/sigmoid, ReLU, Softmax
* input layer, output layer, hidden layers
* weight parameter, bias parameter
* dense layer, flatten layer, max-pooling layer, convolutional layer, dropout layer,
* the gradient descent algorithm, the stochastic gradient descent algorithm
* model underfitting/overfitting, regularization

Ex. 1. What does a flatten layer do?

Ex. 2. For classification tasks, why is it common to use softmax as the activation function in the last layer?

Ex. 3. What is the difference between the gradient descent algorithm and the stochastic gradient descent algorithm?

1. **How to build and train a neural network**

To build:

a). Declare the number of layers,

b). the type of each layer,

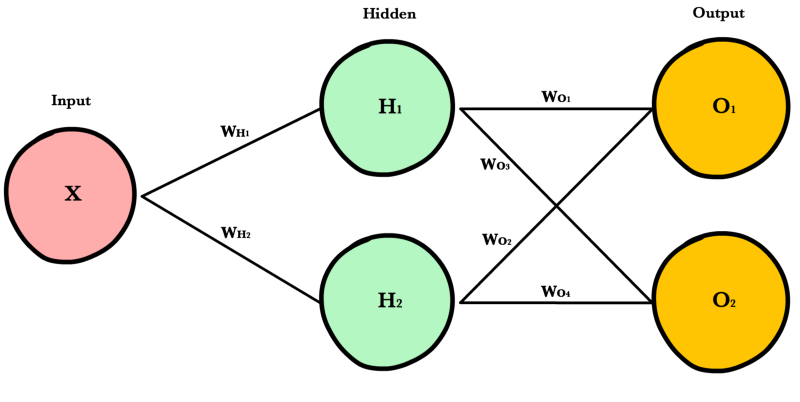
c). the size of each layer,

d). the activation function of each neuron.

To train:

1. Define a cost function / loss function. This function evaluates the model’s performance with current set of parameters. Small cost indicates good performance.
2. Use gradient descent as the training algorithm. Acknowledge that the gradient descent method can end at a local minimum instead of a global minimum.
3. Prevent overfitting: Use the performance on a validation set to detect overfitting. Stop when it is not yet overfitting, or use regularization to induce a simpler model.

Ex. Build the following neural network using TensorFlow:



model = tf.keras.model.Sequential()

model.add(tf.keras.layers.Dense(2, input\_size=1, activation=??)) # hidden layer

model.add(tf.keras.layers.Dense(2, activation=??)) # output layer

model.compile(loss=??, optimizer=”sgd”/”adam”, metrics=[“accuray”])

model.fit(x\_train, y\_train, validation\_split=0.1, epoch=??)

1. **The math related to neural networks**
2. Each input neuron takes a feature value from the input data.
3. For each neuron in a hidden layer, it calculates a weighted sum of the value from the previous layer. This is the input of that neuron.
4. The neuron applies its activation function to the input.
5. The neuron broadcasts the output to the next layer.
6. The output from the output layer is the final output.

Ex. Using the above neural network, calculate the model’s output for x = 2.

We need to know the parameter values.

Suppose all biases are zero.

Weights: [1, 2, 3, -4, 5, -6]

Activation function: ReLU(t) = max{0, t}

Model’s output: (26, 0)

1. **The strengths and weaknesses of neural networks**

- What are neural network’s applications?

large dataset, complicated data(images, audio, video, texts), challenging tasks (things cannot be easily explained: robotics, data generation, group dynamics, text analysis)

- When should we avoid using neural network model?

small dataset, simple data, simple task

sensitive tasks (health care, financial operations)