**Global AI Intern - ML Cheat Sheet Write-up**

Cheng-Yun Tsai

**1. Linear Regression**

Linear Regression can be classified into Simple Regression and Multivariable Regression, where Simple Regression only have one coefficient and one bias and Multivariable Regression has more than one coefficient. Like general ML training process, we need to do feature engineering (Normalization for Multivariable Regression), design a cost function and the learning method. In the Linear Regression section, the author used MSE as its cost function and simple Gradient Descent as its learning method.

**2. Gradient Descent**

Gradient Descent is a general learning method used in ML. The basic concept of Gradient Descent comes from Calculus (Differentiation). The aim of ML algorithms is to minimize the value of the cost function (MSE in the last section). Hence to minimize the cost function, Gradient Descent decrease the value according to the learning rate by using the differential value of the cost function.

**3. Logistic Regression**

Logistic Regression has three types, Binary, Multi and Ordinal. The document implemented the Binary Logistic and used Scikit Learn for Multi Logistic and applied Cross-Entropy as the cost function. The concept of Binary and Multi Logistic are quite similar, they both adopted the concept of sigmoid function and initialize/ estimate the parameters in the function as linear regression.

**4. Glossary**

Explained the terms for ML.

**5. Calculus**

Introduced basic Calculus concepts include derivatives, integrals, chain rule and gradients.

**6. Linear Algebra**

Introduced basic Linear Algebra concepts include vectors, matrixes, and a python package NumPy.

**7. Probability**

No content.

**8. Statistics**

No content.

**9. Notation**

Math notations.

**10. Concepts**

Concepts of NN. NN is composed of neurons and synapses. In mathematics, it can be represented by weights, parameters, and bias in matrix form. Multiplying parameters in a layer with wights, Adding the bias and applying the result to the activation function, it comes out to be the input value of the next layer.

**11 & 12. Forwardpropagation & Backpropagation**

Forward propagation and Backpropagation are technique for updating parameters in NN, that is, a NN learning method. As we described in previous section, we need to update the parameters to minimize the loss function (cost function). However, calculating derivatives of loss function can only generates the value and direction we need to alter (a real number), but not the values (a matrix) we can apply to update our parameters. Hence, we need to calculate derivatives for every parameter from the loss function. Calculating derivatives in a close form is not that convenient for every case, so we can use backpropagation to calculate it. Forward propagation is the technique to store the values that used for calculating derivatives in backpropagation.

**13. Activation Functions**

Activation functions are functions generate outputs of every NN layer. It can also be classified into continuous and discrete functions based on its outputs. Common activation includes ReLU, ELU, LeakyReLU, Sigmoid, Tanh and Softmax. Softmax function transforms values into probabilities (sum of all value equal to one), hence it is commonly used in the final output NN layer.

**14. Layers**

We can combine various layers in an NN model based on the design of the model. Convolution layer from CNN algorithm is frequently used in computer vision. Dropout layer randomly sets activations to zero to avoid model overfitting. LSTM, Pooling and RNN layers are also some common layers.

**15. Loss Functions**

As previously mentioned, the aim of ML algorithm is to maximize the objective function (minimize the loss function). So, we need to choose an appropriate loss function to make the objective of the model accurate. Cross-entropy, MSE, MAE are common loss functions.

**16. Optimizers**

After designing NN, objective function and calculating derivatives, we need to choose a way to update parameters. Optimizers are algorithms for updating parameters, a bad optimizer may let the model be trapped in local minimum and bad accuracy. Adagrad and Adam are commonly used optimizer.

**17. Regularization**

To prevent model from overfitting, we can try to adopt some methods in the model. We can try different methods in data, model, and training process. For data, feature engineering methods such as Data Augmentation and Noise Injection are used for small training dataset. In terms of model, L1 and L2 Regularization can be adopted into the loss function. In training process, setting early stopping might be a good way to avoid overfitting.