

LOSS FUNCTIONS:

Regression:

1. Squared Loss: The outliers change the model noticeably.
2. Absolute Loss: Don't go out of the way for outliers and might lead poor prediction.

Both of these can lead poor model performance.

3. Pseudo-Huber Loss: It reduces the effects of outliers on model while still being differentiable.

Classification:

1. Cross Entropy Loss
2. KL Divergence
3. Hinge Loss: Maximizes the minimum margin. Penalized if in margin during test time

You can do trial and error method to decide which loss function is suitable for your model.
To avoid trial and error method there is Adaptive Loss

OPTIMIZERS:

They define how neural networks learn

Gradient Descent: Taking small steps iteratively and have larger jumps as it updates the weights after seeing whole dataset.

Stochastic Gradient Descent: It updates the weights after seeing each data point and it can make very noisy jumps that can go away from the optimal values.

Mini-Batch Gradient Descent: It updates parameters only after few samples

SGD + Momentum: It learns faster by paying little attention to the few examples that throw it off time to time

SGD + Momentum + Acceleration

Adagrad: An Adaptive Loss

Adadelat

Adam

None of it is best for all model, it depends on the dataset to choose best optimizer for that model