Optimization Odyssey:

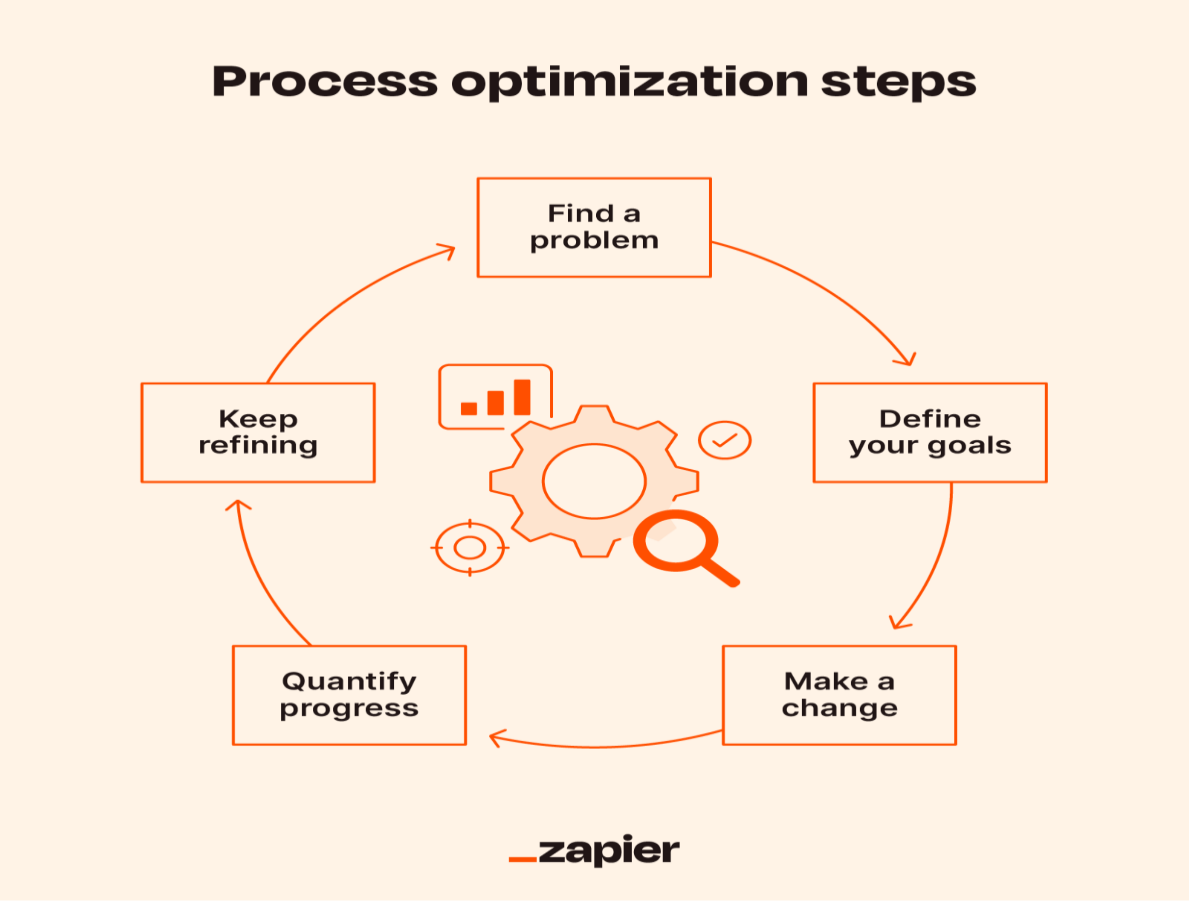
Navigating the Maze of Neural Network Tuning

Revolutionizing Machine Learning: The Power of Artificial Neural Networks

- **Pattern and Speech** recognition.
- **Natural Language** Processing.
- Handling **High-Dimensional data** and learning **Hierarchical representations.**
- Development of **Autonomous vehicles, Robotics.**
- **Democratization of AI.**

The Optimization Phenomenon

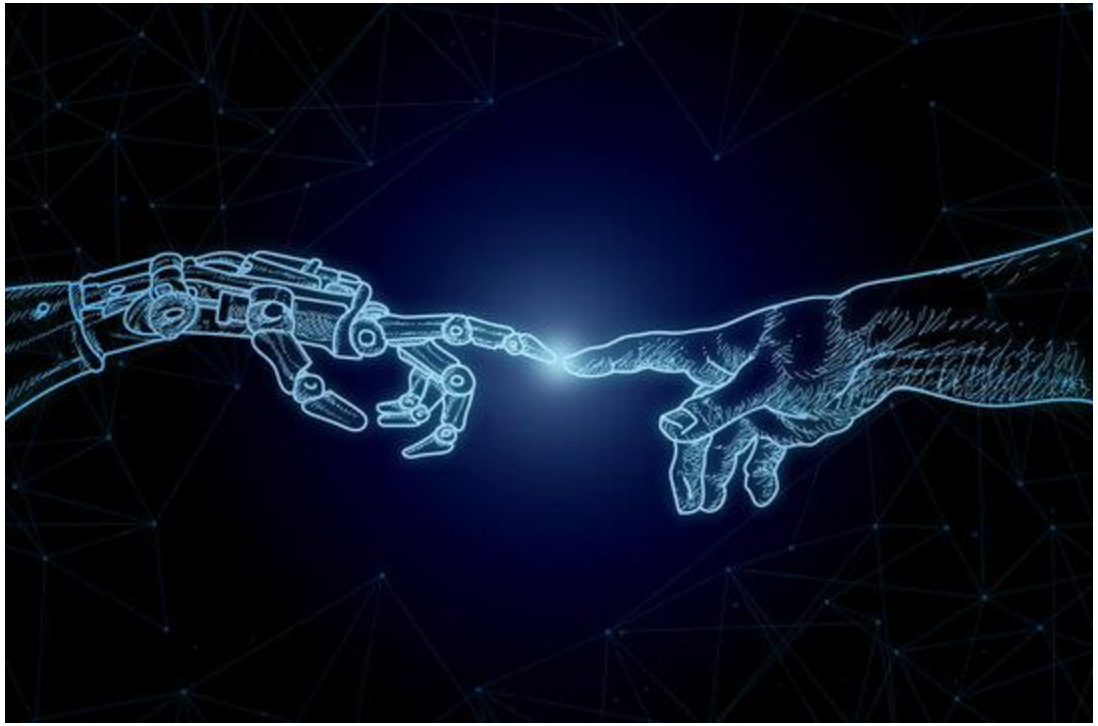
- Problem Formulation
- Iterative process
- Solution evaluation
- Solution refinement
- Solution implementation



Convergence of Optimization and Artificial Biology: Exploring the Synergy of Efficiency and Adaptation

- **Where the human concept of trial and error and the artificial art of continuous learning collide**

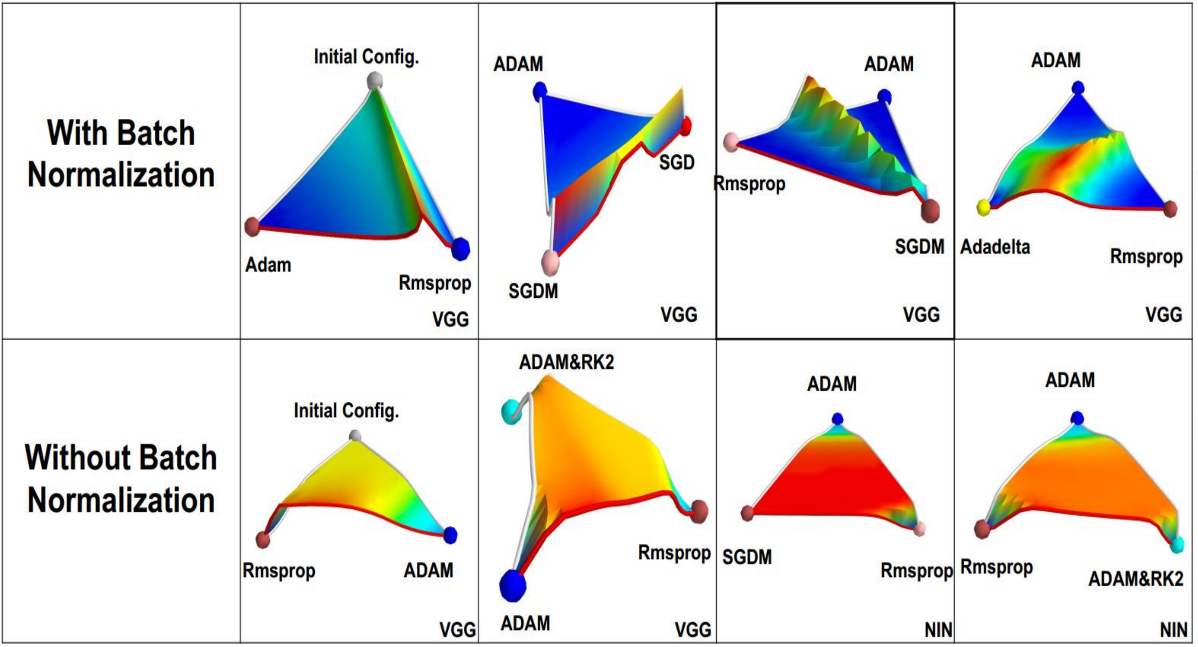
Continuous Learning

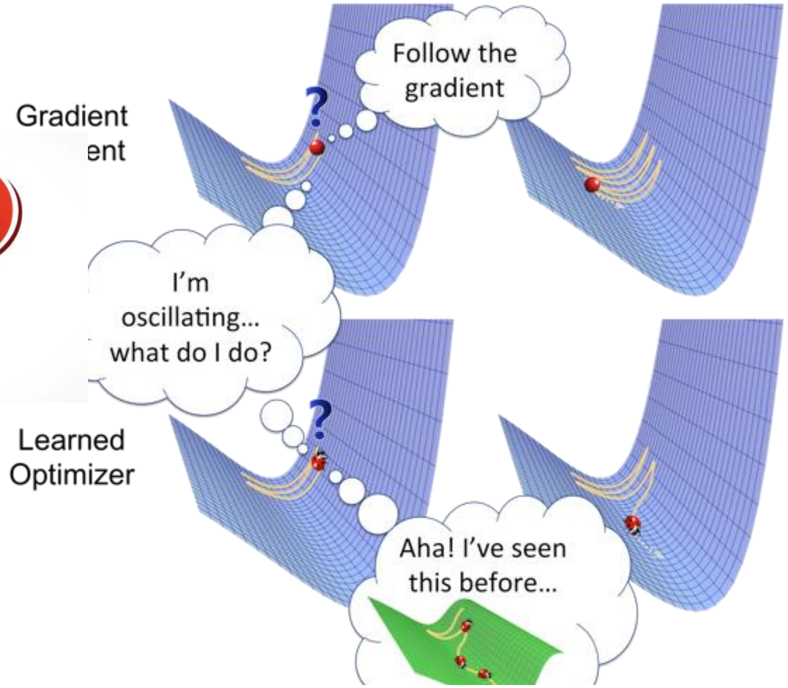


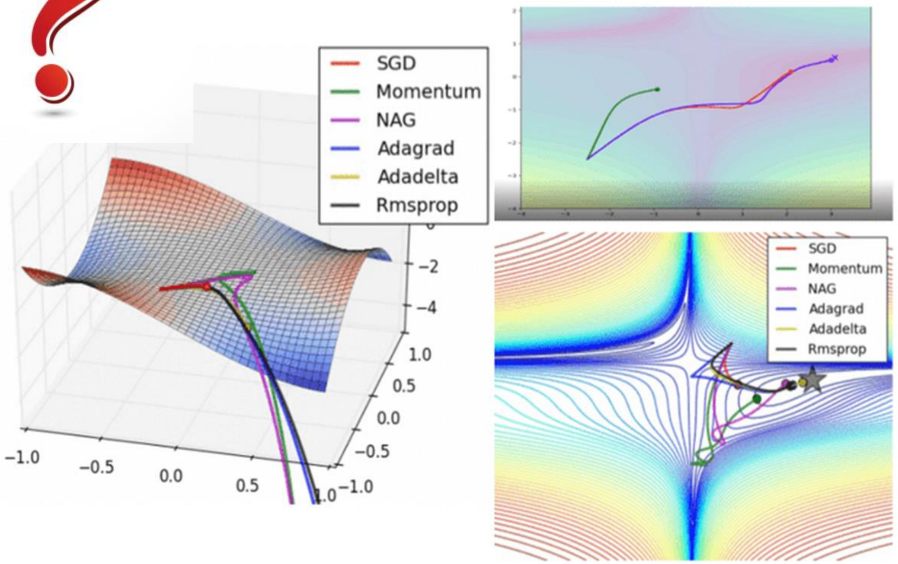
Trial, failure, and the will to try again

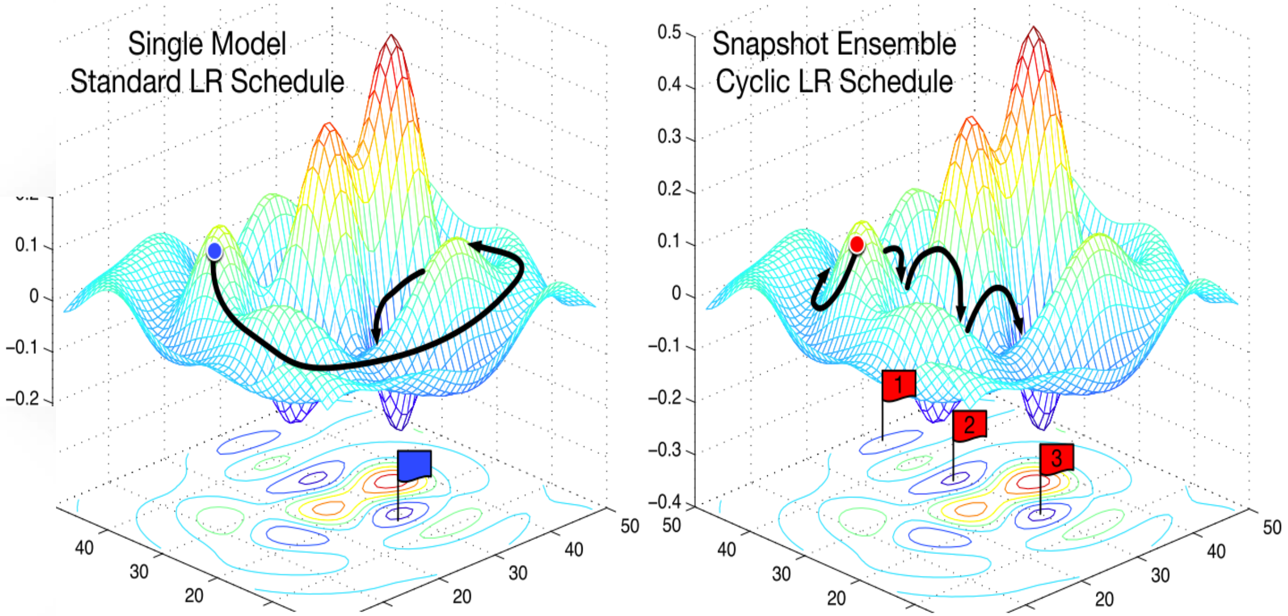
Optimization Frontier: Unlocking Algorithmic Brilliance

- Gradient Descent (GD)
- Stochastic Gradient Descent (SGD)
- Mini-Batch Gradient Descent
- Momentum
- Nesterov Accelerated Gradient (NAG)
- Levenberg-Marquardt
- Quickprop
- Rprop (Resilient Backpropagation)
- Adagrad (Adaptive Gradient)
- RMSProp (Root Mean Square Propagation)
- Adam (Adaptive Moment Estimation)
- AdaDelta (Adaptive Learning Rate)
- Adamax
- Limited-memory BFGS (L-BFGS)
- Nadam (Nesterov-accelerated Adaptive Moment Estimation)
- AMSGrad (Adaptive Moment Estimation with Stochastic Gradient Descent)
- Adadelta
- Natural Evolution Strategies (NES)
- Evolution Strategies (ES)
- Conjugate Gradient (CG)


With Batch Normalization
Initial Config.
Adam
Rmsprop
VGG
ADAM
SGD
SGDM
VGG
ADAM
Rmsprop
SGDM
VGG
ADAM
Adadelta
Rmsprop
VGG
Without Batch Normalization
Initial Config.
Rmsprop
ADAM
VGG
ADAM&RK2
Rmsprop
ADAM
VGG
ADAM
SGDM
Rmsprop
NIN
ADAM
Rmsprop
ADAM&RK2
NIN


Gradient
ent
Follow the gradient
I'm oscillating... what do I do?
Learned Optimizer
Aha! I've seen this before...


SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop


Single Model
Standard LR Schedule
Snapshot Ensemble
Cyclic LR Schedule
1
2
3

In Pursuit of Perfection: The Big 5

A Cut Above the Rest: Introducing the 5 Superior Optimizer Algorithms for Our Project

- ***SGD (Stochastic Gradient Descent)***
- ***Adam (Adaptive Moment Estimation)***
- ***Adagrad (Adaptive Gradient Algorithm)***
- ***RMSProp (Root Mean Square Propagation)***
- ***Nadam (Nesterov-accelerated Adaptive Moment Estimation)***

Breast Cancer Classification with different Optimization Algorithms

- ❑ Which one of the big 5 is more optimal in terms of Accuracy and Loss Function?

Let us find out..

Importing Libraries

```
In [ ]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        import sklearn.datasets
        from sklearn.model_selection import train_test_split
```

Data Collection and Processing

```
In [ ]: # Loading the data from sk.learn
        breast_cancer_dataset =  sklearn.datasets.load_breast_cancer()
```

```
In [ ]: # Loading the data to a Panda Dataframe
        df = pd.DataFrame(breast_cancer_dataset.data, columns = breast_cancer_dataset.feature_names)
```

Standardize the Data (Normalising for better accuracy)

```
In [ ]: from sklearn.preprocessing import StandardScaler
```

```
In [ ]: scaler = StandardScaler()

        X_train_std = scaler.fit_transform(X_train)

        X_test_std = scaler.transform(X_test)
```

```
In [ ]: print(X_train_std)
```

Building The Neural Network

```
In [ ]: # Importing tenserflow and Keras
        import tensorflow as tf
        tf.random.set_seed(3)  #for constant accuracy score in more than one session
        from tensorflow import keras
```

Adam Model

In [ ]:
```
# setting up the layers of NN
# flatten is taking the 30 feautures and putting it into 1 dimensional array
# first dense is hidden, second is output layer

adam= keras.Sequential([
                        keras.layers.Flatten(input_shape=(30,)),
                        keras.layers.Dense(20, activation='relu'),
                        keras.layers.Dense(2, activation='sigmoid')
])
```

In [ ]:
```
# Compiling NN

adam.compile(optimizer='adam',
             loss='sparse_categorical_crossentropy',
             metrics=['accuracy'])
```

In [ ]:
```
# training the NN

history_adam = adam.fit(X_train_std, Y_train, validation_split=0.1, epochs=10)
```

Visualing accuracy and loss

In [ ]:
```
plt.plot(history_adam.history['accuracy'])
plt.plot(history_adam.history['val_accuracy'])

plt.title('model_accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')

plt.legend(['Training Data', 'Validation Data'], loc = 'lower right')
```

In [ ]:
```
plt.plot(history_adam.history['loss'])
plt.plot(history_adam.history['val_loss'])

plt.title('model_loss')
plt.ylabel('loss')
plt.xlabel('epoch')

plt.legend(['Training Data', 'Validation Data'], loc = 'upper right')
```

Accuracy of the model on the test data

In [ ]:
```
loss, accuracy = adam.evaluate(X_test_std, Y_test)
print(accuracy)
```

SGD model

In [ ]:
```
sgd = keras.Sequential([
                        keras.layers.Flatten(input_shape=(30,)),
                        keras.layers.Dense(20, activation='relu'),
                        keras.layers.Dense(2, activation='sigmoid')
])

sgd.compile(optimizer='SGD',
             loss='sparse_categorical_crossentropy',
             metrics=['accuracy'])
```

In [ ]:
```
history_sgd = sgd.fit(X_train_std, Y_train, validation_split=0.1, epochs=10)
```

In [ ]:
```
loss, accuracy = sgd.evaluate(X_test_std, Y_test)
print(accuracy)
```

RMSProp

In [ ]:
```
rms = keras.Sequential([
                        keras.layers.Flatten(input_shape=(30,)),
                        keras.layers.Dense(20, activation='relu'),
                        keras.layers.Dense(2, activation='sigmoid')
])

rms.compile(optimizer='RMSprop',
             loss='sparse_categorical_crossentropy',
             metrics=['accuracy'])
```

In [ ]:
```
history_rms = rms.fit(X_train_std, Y_train, validation_split=0.1, epochs=10)
```

In [ ]:
```
loss, accuracy = rms.evaluate(X_test_std, Y_test)
print(accuracy)
```

Adagrad

```
In [ ]: adagra = keras.Sequential([
                            keras.layers.Flatten(input_shape=(30,)),
                            keras.layers.Dense(20, activation='relu'),
                            keras.layers.Dense(2, activation='sigmoid')
        ])

        adagra.compile(optimizer='Adagrad',
                     loss='sparse_categorical_crossentropy',
                     metrics=['accuracy'])
```

```
In [ ]: history_adagra = adagra.fit(X_train_std, Y_train, validation_split=0.1, epochs=10)
```

```
In [ ]: loss, accuracy = adagra.evaluate(X_test_std, Y_test)
        print(accuracy)
```

Nadam

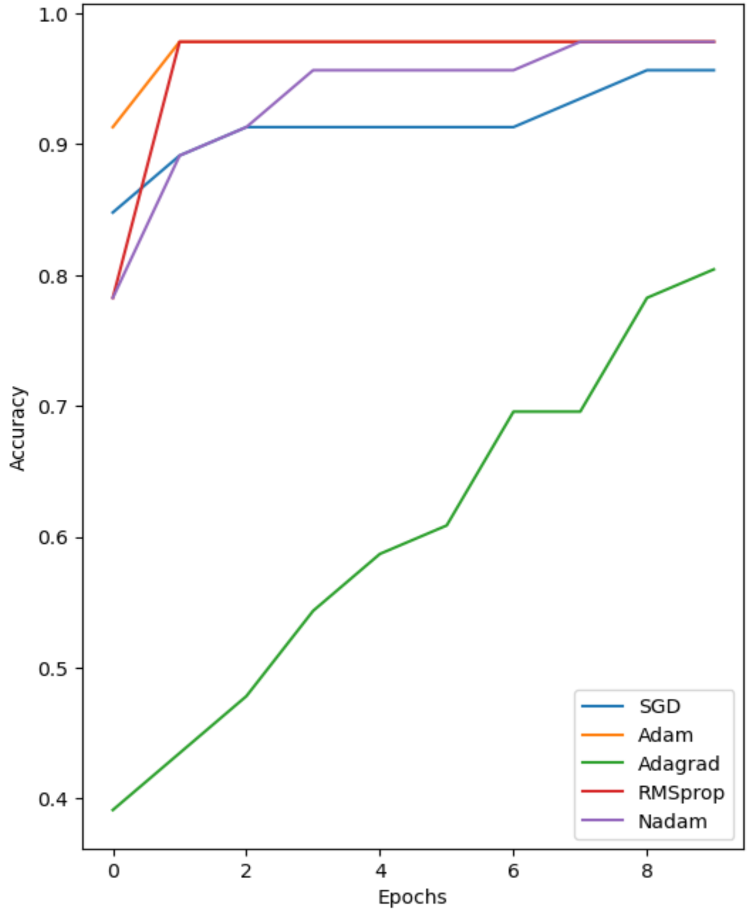
```
In [ ]: nadam = keras.Sequential([
                            keras.layers.Flatten(input_shape=(30,)),
                            keras.layers.Dense(20, activation='relu'),
                            keras.layers.Dense(2, activation='sigmoid')
        ])

        nadam.compile(optimizer='Nadam',
                     loss='sparse_categorical_crossentropy',
                     metrics=['accuracy'])
```

```
In [ ]: history_nadam = nadam.fit(X_train_std, Y_train, validation_split=0.1, epochs=10)
```

```
In [ ]: loss, accuracy = nadam.evaluate(X_test_std, Y_test)
        print(accuracy)
```

Findings



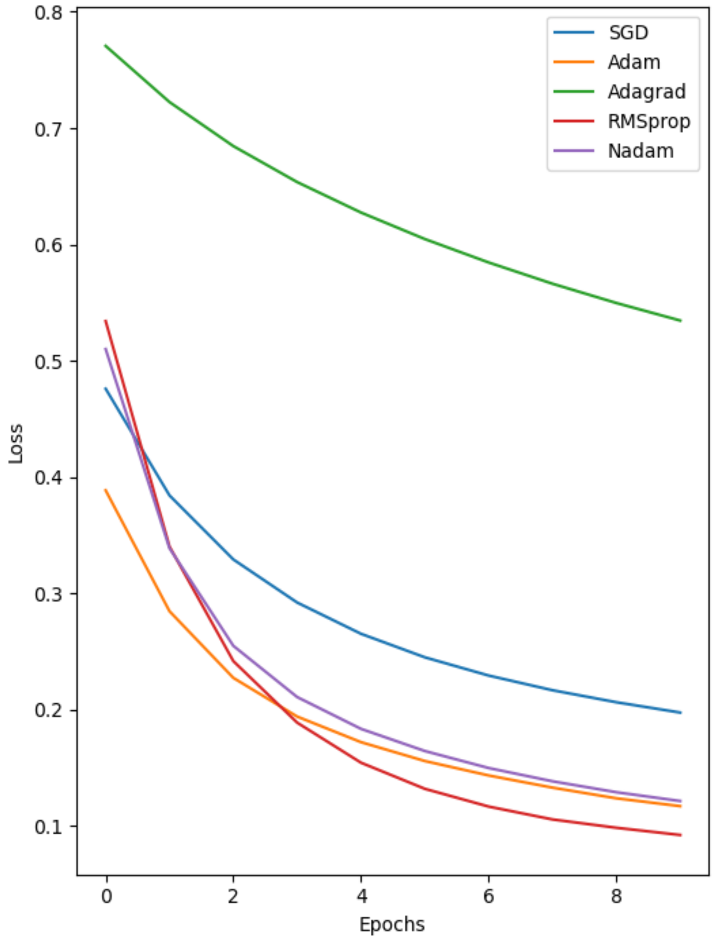
Behavior in Accuracy

B. Model Evaluation

The model is evaluated using accuracy and loss function, briefly structured below.

Algorithm	Accuracy	Loss Function
SGD	95.61%	0.1669
Adam	97.37%	0.1273
Adagrad	70.18%	0.6419
RMSProp	95.61%	0.1276
Nadam	95.61%	0.1389

Table 1: Evaluation of the algorithms



Behavior in Loss Function

Adam takes the ~~apple~~ fruit

With the highest Accuracy of rate 97.37% *and* the lowest Loss Function of 0.1273, the Adam optimizer is clearly the most powerful optimizer in this study.

Slapping the rest out of *The Office*

- **Get outta here, Chris Rock, sorry, you pretender algorithms..**

End of the voyage through ANN with optimizers

Riashad Hassan
ID: 20320001
Department of CSE
Fareast International University

Shahidul Islam
ID: 0722210005101007
Department of CSE
Fareast International University

Shabab Affanul Hoque
ID: 0722210005101003
Department of CSE
Fareast International University