

M3402



MyAlgo



MyAlgo

MyAlgo is my Machine Learning project which is focused on building Linear Regression and Artificial Neural Network from scratch without using high level machine learning libraries such as Tensorflow, PyTorch and scikit-learn.

The main goal of this project is to combining

- **python** as a user facing python API
- **C++** as a high performance computational backend (only for numerical calculations)

Implemented Algorithms

Linear Regression

Feature	Supported values
Methods	Normal Equation, GD, SGD
Regularization	Lasso, Ridge, Elastic net

Example usage:

```
model = LinearRegression(method='gd', lr=0.01, epochs=1000)
model.fit(X_train, y_train)
predictions = model.predict(X_test)
```



2. ANN

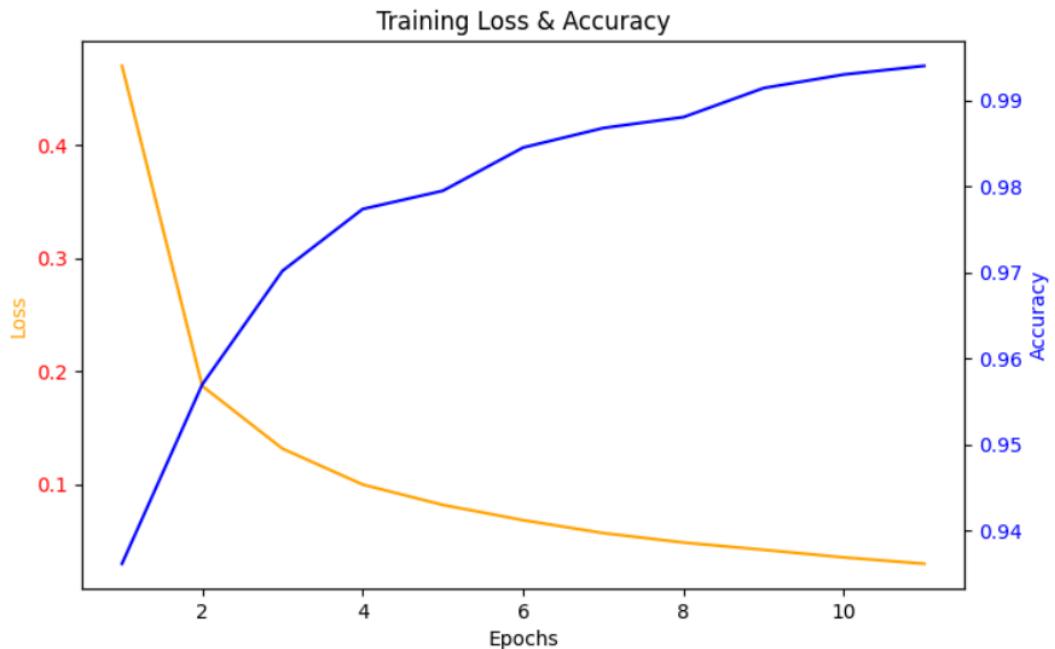
Feature	Supported values
Layers	Flatten, Dense
Activations	ReLU, sigmoid, Tanh, Softmax
Loss functions	MSE, cross-entropy

Example usage:

```
model = NeuralNetwork(
    Flatten((28, 28)),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
)
model.compile(loss='categorical_crossentropy', lr=0.01)
model.train(X_train, y_train, epochs=11, batch_size=32)
```



Results of MNIST dataset training

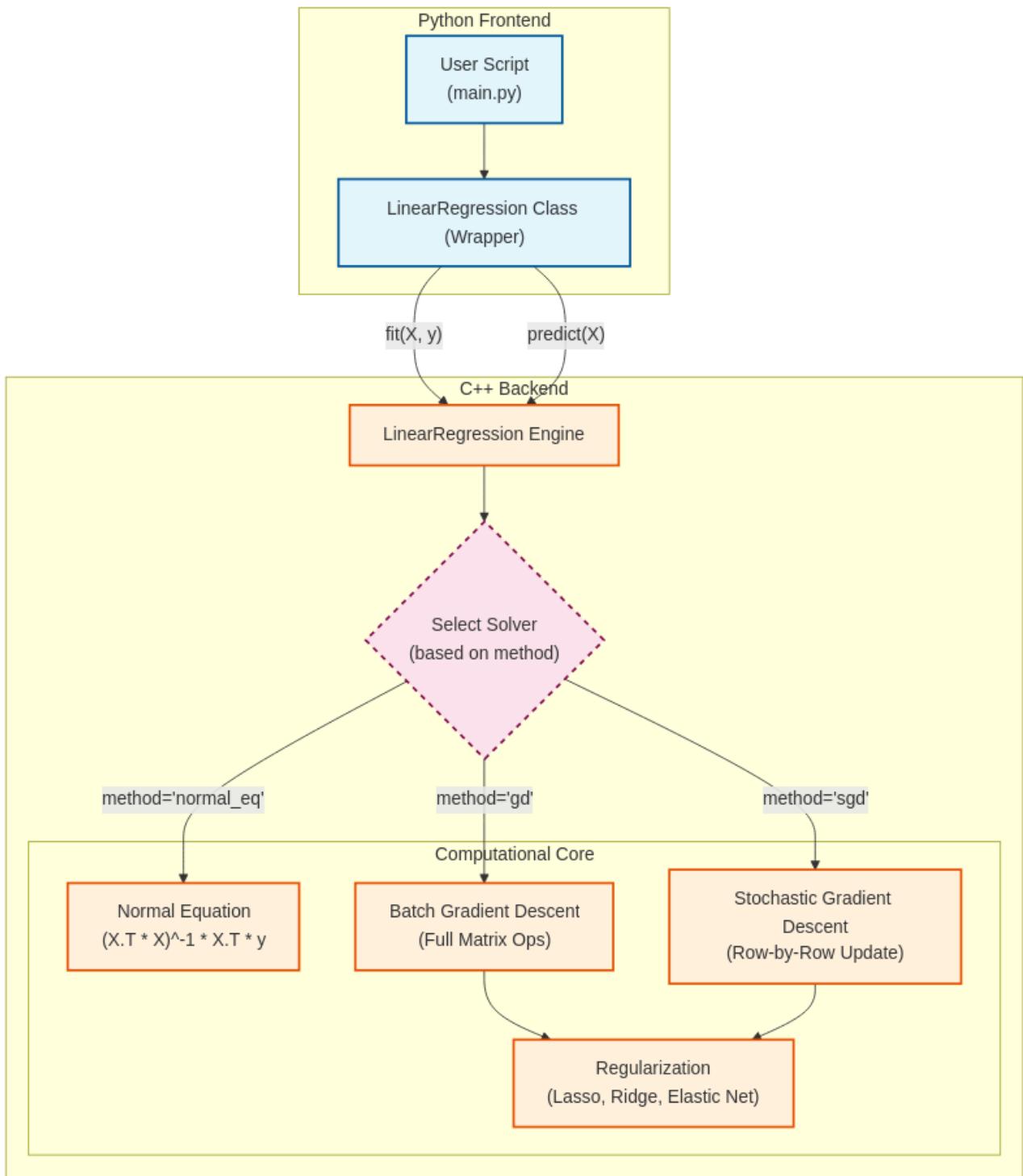


Result achieved on the MNIST dataset with 80% training and 20% testing set.

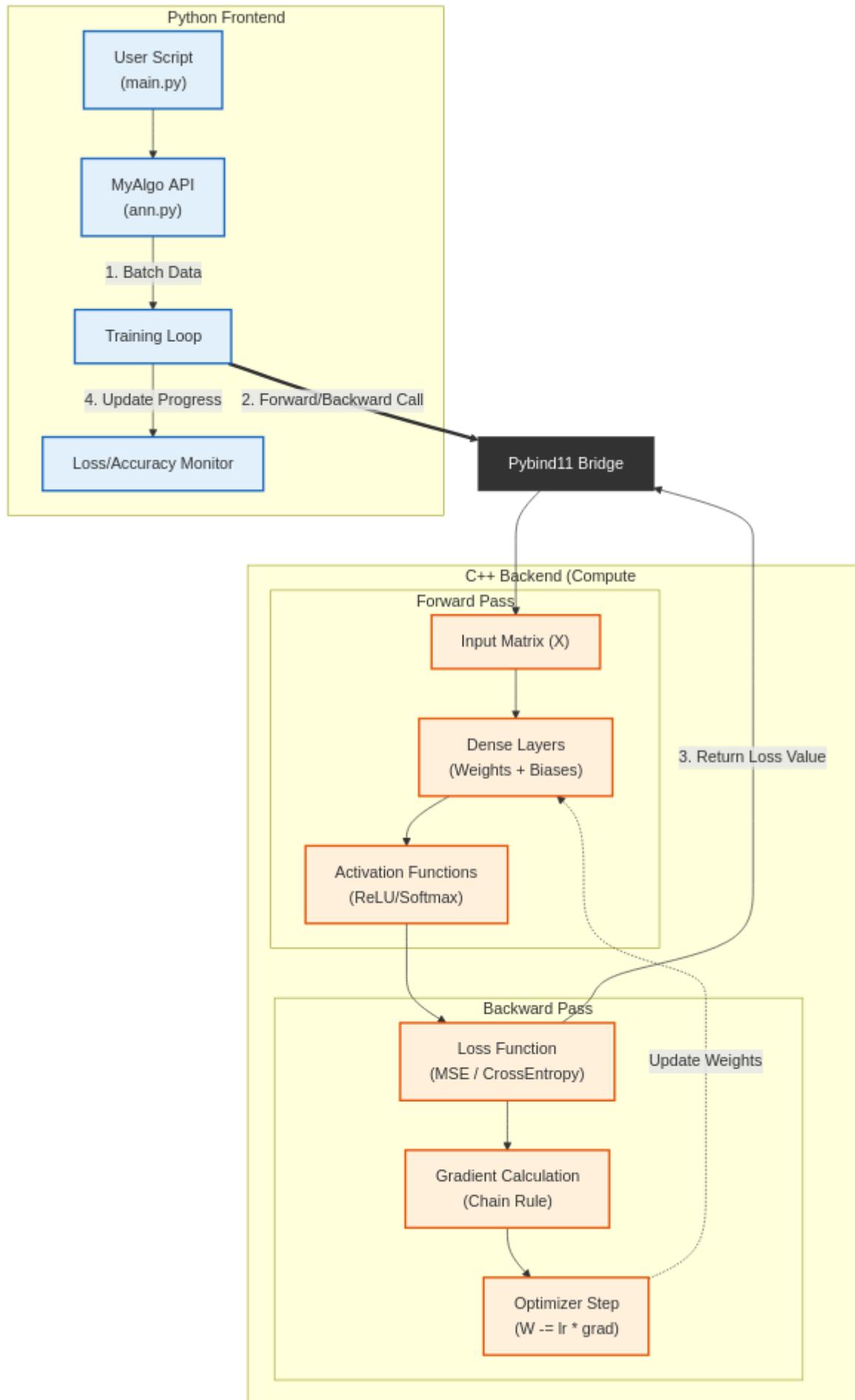
```
(.venv) vitthalhumbe@fedora:~/Projects/MyAlgo$ python3 ann/ann.py
Epoch 1/11
100%|██████████| 875/875 , - loss=0.4699052150826165
Epoch 2/11
100%|██████████| 875/875 , - loss=0.18718145482428133
Epoch 3/11
100%|██████████| 875/875 , - loss=0.13178852348946632
Epoch 4/11
100%|██████████| 875/875 , - loss=0.10008366357350704
Epoch 5/11
100%|██████████| 875/875 , - loss=0.08207424477302873
Epoch 6/11
100%|██████████| 875/875 , - loss=0.06858756376078465
Epoch 7/11
100%|██████████| 875/875 , - loss=0.05719224766952203
Epoch 8/11
100%|██████████| 875/875 , - loss=0.04883338036107661
Epoch 9/11
100%|██████████| 875/875 , - loss=0.04239642393108091
Epoch 10/11
100%|██████████| 875/875 , - loss=0.03569645383758586
Epoch 11/11
100%|██████████| 875/875 , - loss=0.030127117759553318

Final Test Accuracy: 0.9751428571428571
Final Test loss: 0.08576073052489953
```

Architecture of Linear Regression class, and the dataflow for training the Linear Regression model with Multiple training methods.



ANN Architecture, with a training loop handled by the python API and the C++ is used only for matrix multiplication, weight changing, activation functions, etc. mathematical operations



THANK YOU