Laporan Tugas Besar 1
IF3270 Pembelajaran Mesin

Feed Forward Neural Network (FFNN)

Dipersiapkan oleh:

1. Rici Trisna Putra 13522026
2. Imam Hanif Mulyarahman 13522030
3. Diero Arga Purnama 13522056

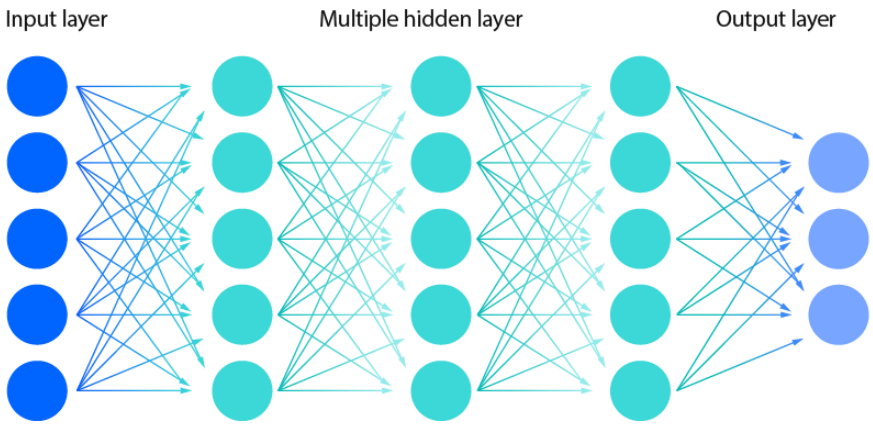
Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung,
Jl. Ganesha 10, Banding 40132

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
DESKRIPSI PERSOALAN



Gambar 1.1 Feed Forward Neural Network (FFNN)

Pembelajaran Mesin (Machine Learning) merupakan cabang ilmu komputer yang berfokus pada pengembangan algoritma yang mampu belajar dari data untuk membuat keputusan. Salah satu metode yang populer dalam pembelajaran mesin adalah Feedforward Neural Network (FFNN). FFNN termasuk dalam Artificial Neuron Network (ANN).

FFNN memiliki arsitektur berlapis yang mengalirkan data secara maju (forward) dari lapisan input ke lapisan output melalui beberapa lapisan tersembunyi (hidden layer). Pada setiap lapisan, neuron menerima masukan dari neuron pada lapisan sebelumnya, melakukan perhitungan berdasarkan bobot tertentu, dan kemudian menerapkan fungsi aktivasi untuk menghasilkan keluaran yang akan digunakan oleh lapisan berikutnya. Proses ini dikenal sebagai forward propagation.

Selain itu, FFNN juga melakukan proses yang disebut backward propagation. Proses ini bertujuan untuk meminimalkan galat (error) dengan menyesuaikan bobot jaringan berdasarkan perbedaan antara hasil prediksi dan nilai aktual. Pembaruan bobot ini dilakukan melalui algoritma optimasi seperti gradient descent, yang memperhitungkan turunan (gradient) dari fungsi galat terhadap bobot untuk menemukan nilai optimal yang meminimalkan galat tersebut.

Pada tugas besar ini, kami diminta untuk membuat model FFNN dari dasar (from scratch). Tugas ini mencakup bagian inisialisasi bobot, proses forward propagation, perhitungan galat (error), backward propagation, hingga pembaruan bobot. Hal ini bertujuan agar memiliki pemahaman yang kuat dan mendalam mengenai mekanisme kerja FFNN.

BAB II
PEMBAHASAN

2.1. Penjelasan Implementasi

2.1.1. Deskripsi Kelas

a. Neuron

Kelas Neuron adalah kelas yang merepresentasikan neuron pada FFNN. Adapun penjelasan atribut dan methodnya sebagai berikut.

Tabel 2.1.1.1 Atribut Kelas Neuron

Atribut	Deskripsi
weights	Bobot dari neuron
weights_gradients	Gradien dari bobot neuron
bias	Bobot bias dari neuron
bias_gradients	Gradien dari bobot bias neuron
value_matrice	Nilai suatu neuron dalam bentuk array untuk menangani kasus batch

Tabel 2.1.1.2 Method Kelas Neuron

Method	Deskripsi
init	Inisialisasi kelas neuron

b. Layer

Kelas layer adalah kelas yang merepresentasikan layer pada FFNN. Adapun penjelasan atribut dan methodnya sebagai berikut.

Tabel 2.1.1.3 Atribut Kelas Layer

Atribut	Deskripsi
n_neurons	Jumlah neuron dalam layer
activation_func	Fungsi aktivasi layer
neurons	Neuron yang ada dalam layer
weight_matrice	Matriks berisi bobot pada layer

bias_matrice	Matriks berisi bias pada layer
weight_gradients	Matriks berisi gradien bobot pada layer
bias_gradients	Matriks berisi gradien bias pada layer
value_matrice	Matriks berisi nilai pada layer

Tabel 2.1.1.4 Method Kelas Layer

Method	Deskripsi
init	Inisialisasi kelas layer

c. FungsiAktivasi

Kelas Neuron adalah kelas yang merepresentasikan fungsi aktivasi pada FFNN. Adapun penjelasan atribut dan methodnya sebagai berikut.

Tabel 2.1.1.5 Atribut Kelas FungsiAktivasi

Atribut	Deskripsi
name	Nama fungsi aktivasi
func	Memanggil fungsi aktivasi
derivative	Memanggil fungsi turunan aktivasi

Tabel 2.1.1.6 Method Kelas FungsiAktivasi

Method	Deskripsi
linear	Fungsi aktivasi linear
linear_derivative	Fungsi turunan aktivasi linear
relu	Fungsi aktivasi relu
relu_derivative	Fungsi turunan aktivasi relu
sigmoid	Fungsi aktivasi sigmoid
sigmoid_derivative	Fungsi turunan aktivasi sigmoid
tanh	Fungsi aktivasi tanh

tanh_derivative	Fungsi turunan aktivasi tanh
softmax	Fungsi aktivasi softmax
softmax_derivative	Fungsi turunan aktivasi softmax

d. FungsiLoss

Kelas Neuron adalah kelas yang merepresentasikan fungsi loss pada FFNN. Adapun penjelasan atribut dan methodnya sebagai berikut.

Tabel 2.1.1.7 Atribut Kelas FungsiLoss

Atribut	Deskripsi
name	Nama fungsi loss
func	Pemanggil fungsi loss
derivative	Pemanggil fungsi turunan loss

Tabel 2.1.1.8 Method Kelas FungsiLoss

Method	Deskripsi
mse	Fungsi loss Mean Squared Error
mse_derivative	Fungsi turunan loss Mean Squared Error
bce	Fungsi loss Binary Cross-Entropy
bce_derivative	Fungsi turunan loss Binary Cross-Entropy
cce	Fungsi loss Categorical Cross-Entropy
cce_derivative	Fungsi turunan loss Categorical Cross-Entropy
derivative	Memanggil fungsi turunan loss

e. FFNN

Kelas ffnn adalah kelas yang merepresentasikan FFNN. Adapun penjelasan atribut dan methodnya sebagai berikut.

Tabel 2.1.1.9 Atribut Kelas FFNN

Atribut	Deskripsi
layers	Array dari layers
activation_func	Array dari fungsi aktivasi untuk tiap layer
batch_size	Ukuran input
lost_function	Fungsi loss yang dipilih
target	Output yang sebenarnya

Tabel 2.1.1.9 Method Kelas FFNN

Method	Deskripsi
generate_matrices	Membuat matriks untuk atribut layer
update_neurons_from_matrices	Memperbaharui neuron berdasarkan matriks pada layer
forward_propagation	Melakukan forward propagation dan mengembalikan nilai error
backward_propagation	Melakukan backward propagation
train	Melatih model
predict	Melakukan prediksi nilai
save	Melakukan penyimpanan model
load	Melakukan pemuatan model
display_matrices	Menampilkan matriks

f. Visualize

File ini berisi fungsi dan prosedur yang dibutuhkan untuk membuat visualisasi berbentuk graf dari model FFNN yang dibuat. Adapun Fungsi dan prosedur yang ada sebagai berikut.

Tabel 2.1.1.10 Fungsi dan Prosedur Visualisasi

Fungsi dan Prosedur	Deskripsi

visualize_network	Visualisasi FFNN dalam bentuk graf
plot_weight_dist	Membuat plot distribusi bobot
plot_gradient_dist	Membuat plot distribusi gradien bobot

2.1.2. Forward Propagation

Forward propagation adalah proses meneruskan data dari input layer melewati tiap neuron pada hidden layer sampai kepada output layer yang nanti akan dihitung errornya. Pada setiap layer yang dilewati akan dilakukan perhitungan untuk mendapatkan nilai dari neuron yang dituju. Nilai ini diambil dari kombinasi linear dari nilai di lapisan sebelumnya dengan bobot yang akan dimasukkan ke dalam fungsi aktivasi. Perhitungan ini diteruskan sampai layer terakhir yaitu output layer. Hasil dari perhitungan tersebut akan dibandingkan dengan nilai yang diprediksi untuk mendapatkan nilai errornya dari fungsi loss yang diinginkan.

2.1.3. Backward Propagation dan Weight Update

Backward propagation adalah proses untuk meneruskan nilai error yang dihasilkan dari forward propagation kembali ke lapisan awal. Proses ini memperbaharui nilai bobot tiap neuron dan bias dengan learning rate tertentu. Hal ini dilakukan untuk meminimalkan nilai error yang didapatkan pada forward propagation. Proses weight update ini dilakukan menggunakan algoritma Stochastic Gradient Descent. Algoritma ini akan menghitung gradient (turunan) dari loss function yang selanjutnya diteruskan menggunakan metode chain rule hingga mencapai input layer. Adapun rumus dalam melakukan weight update adalah sebagai berikut.

$$w_{ji} = w_{ji} + \Delta w_{ji} = w_{ji} - \eta \frac{\delta E_d}{\delta w_{ji}} = w_{ji} + \eta \delta_j x_{ji}$$

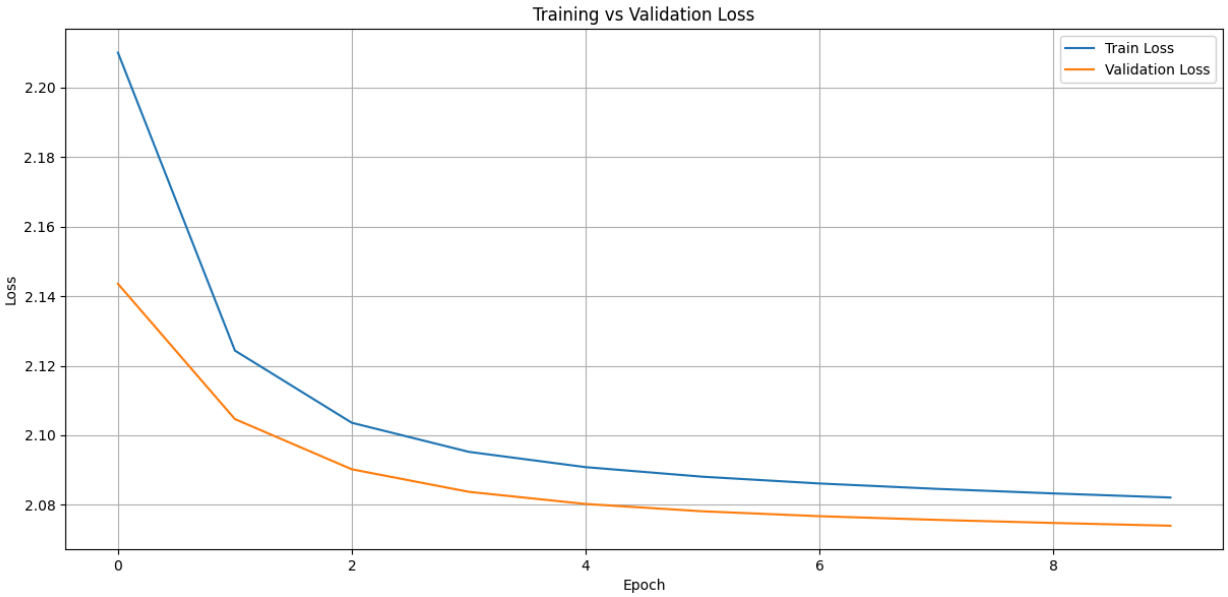
2.2. Hasil Pengujian

2.2.1. Pengaruh depth dan width

Pada bagian ini akan diuji pengaruh dari tingkat kedalaman dan ukuran lebar dari layer terhadap hasil akurasi FFNN.

a. Depth = 1 dan Width = 1

```
Train accuracy: 0.2092
Test accuracy: 0.1994
```

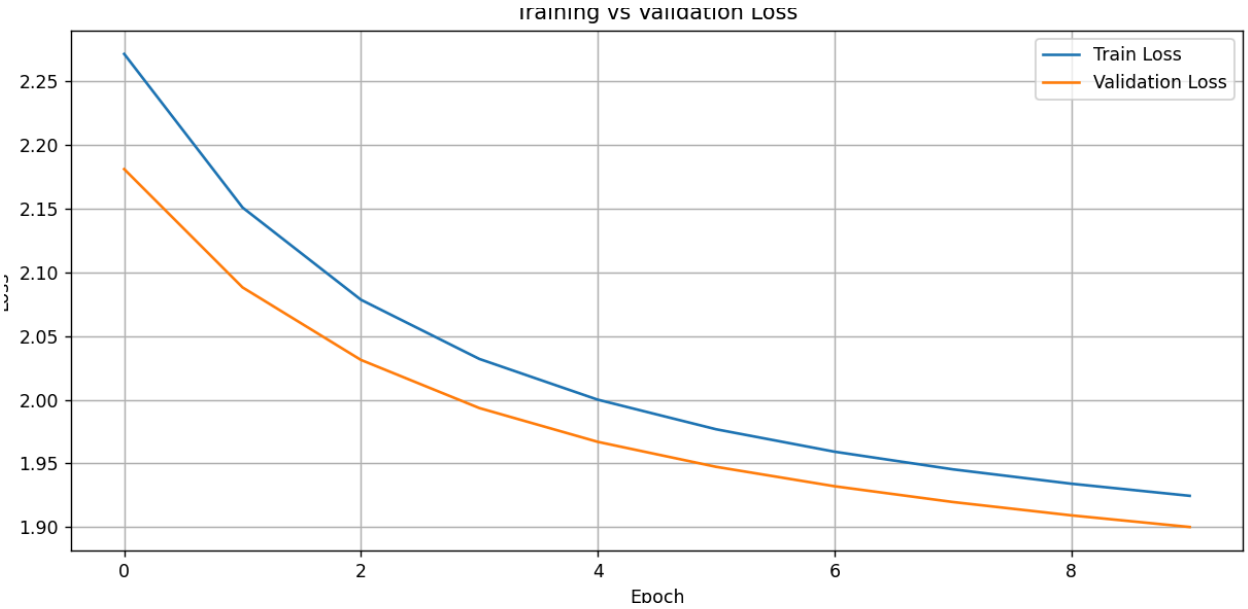


b. Depth = 1 dan Width = 2

```
Train accuracy: 0.2135
Test accuracy: 0.2079
```

c. Depth = 1 dan Width = 3

```
Train accuracy: 0.4223
Test accuracy: 0.4227
```



d. Depth = 2 dan Width = 2


```
Train accuracy: 0.1918
Test accuracy: 0.1805
```

e. Depth = 3 dan Width = 2

```
Train accuracy: 0.1225
Test accuracy: 0.1297
```

f. Depth = 4 dan Width = 2

```
Train accuracy: 0.1335
Test accuracy: 0.1223
```

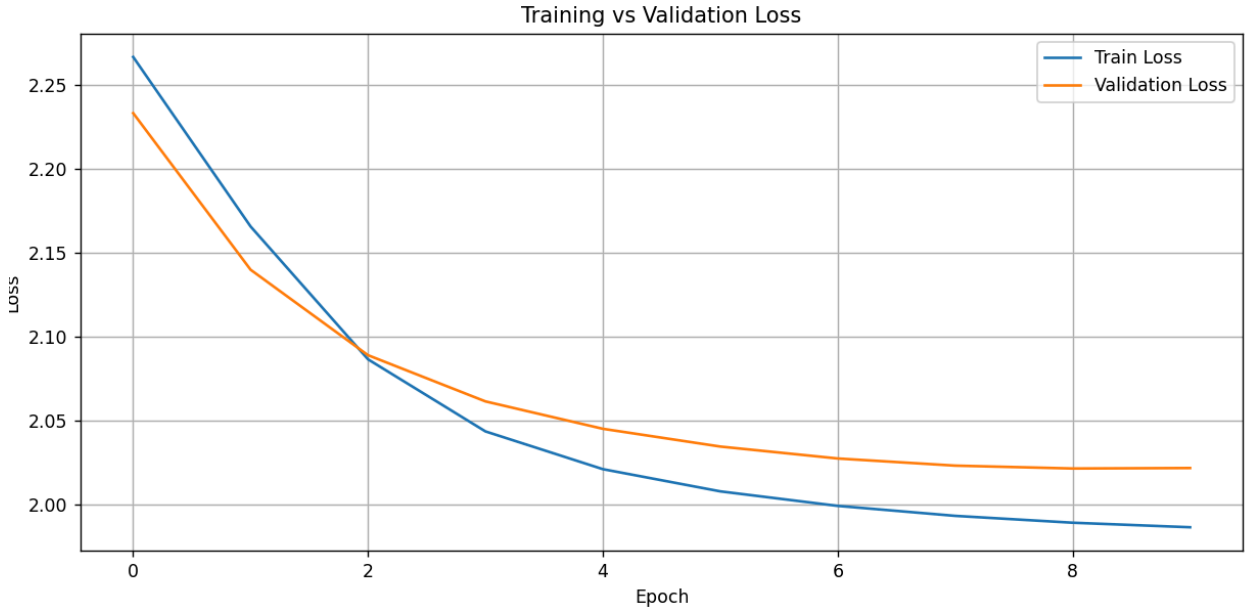
Dari grafik perbandingan diatas, dapat dilihat bahwa tingkat kedalaman dan lebar layer berpengaruh terhadap akurasi model. Model dengan tingkat kedalaman dan lebar yang lebih tinggi tidak menjamin meningkatkan akurasi yang dihasilkan.

2.2.2. Pengaruh fungsi aktivasi

Seluruh percobaan dilakukan dengan depth = 4, width = 2, dan max_epoch = 10.

a. Linear

```
Train accuracy: 0.2980
Test accuracy: 0.1123
```



b. ReLu

```
Train accuracy: 0.2315
Test accuracy: 0.1505
```

c. Sigmoid

```
Train accuracy: 0.2255
Test accuracy: 0.1448
```

d. Hyperbolic Tangent (tanh)

```
Train accuracy: 0.2065
Test accuracy: 0.1952
```

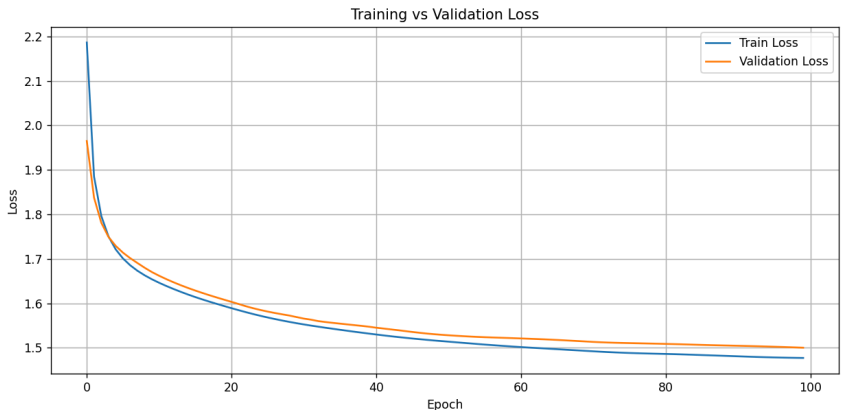
Dapat dilihat bahwa perbedaan fungsi aktivasi berpengaruh besar dalam tingkat akurasi suatu model. Beberapa jenis data lebih sesuai digunakan pada fungsi aktivasi satu dibandingkan dengan yang lainnya.

2.2.3. Pengaruh learning rate

Untuk pengujian menggunakan dataset mnist_784 dengan fungsi aktivasi tanh, ukuran batch 10, lebar neuron 3, kedalaman neuron 1, dan jumlah epoch maksimal 100.

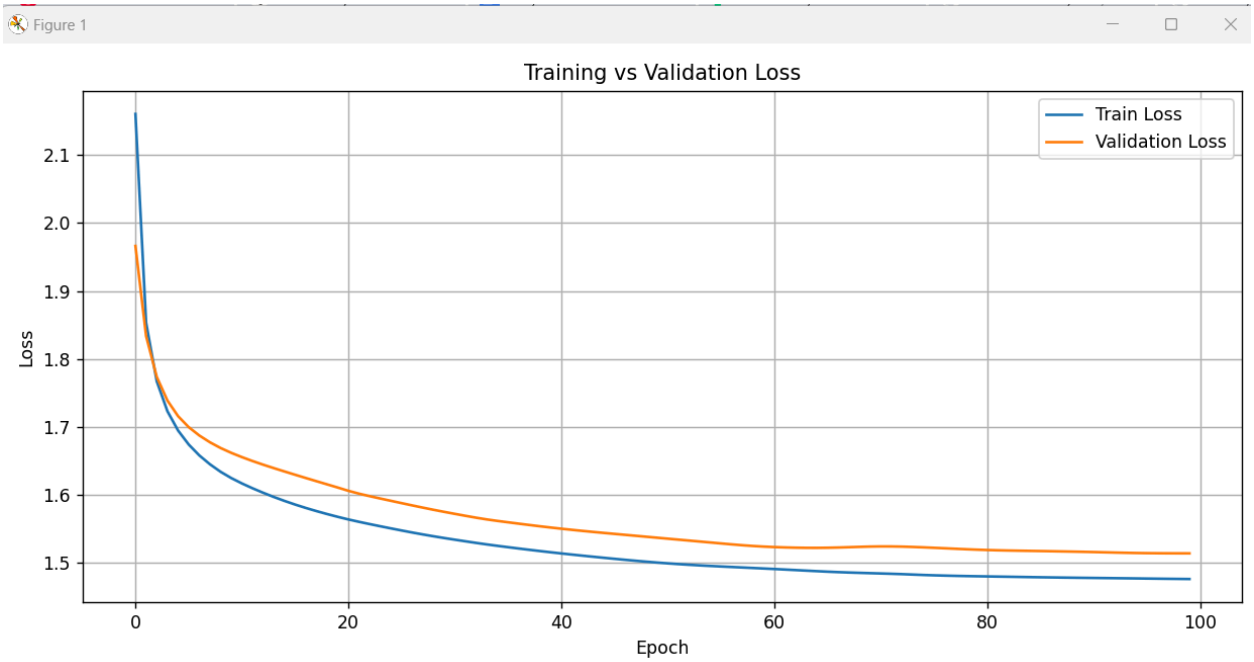
a. Learning rate 0.1

```
Train accuracy: 0.4783
Test accuracy: 0.4784
```



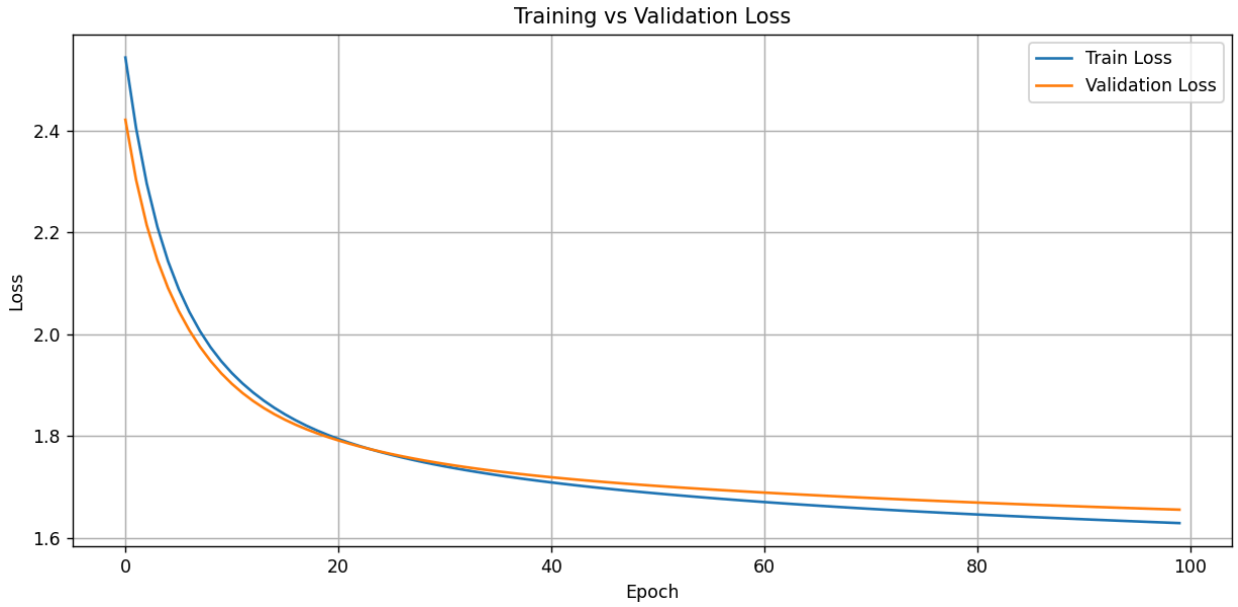
b. Learning rate 0.01

```
Train accuracy: 0.4880
Test accuracy: 0.4665
```



c. Learning rate 0.001

```
Train accuracy: 0.5210
Test accuracy: 0.4956
```

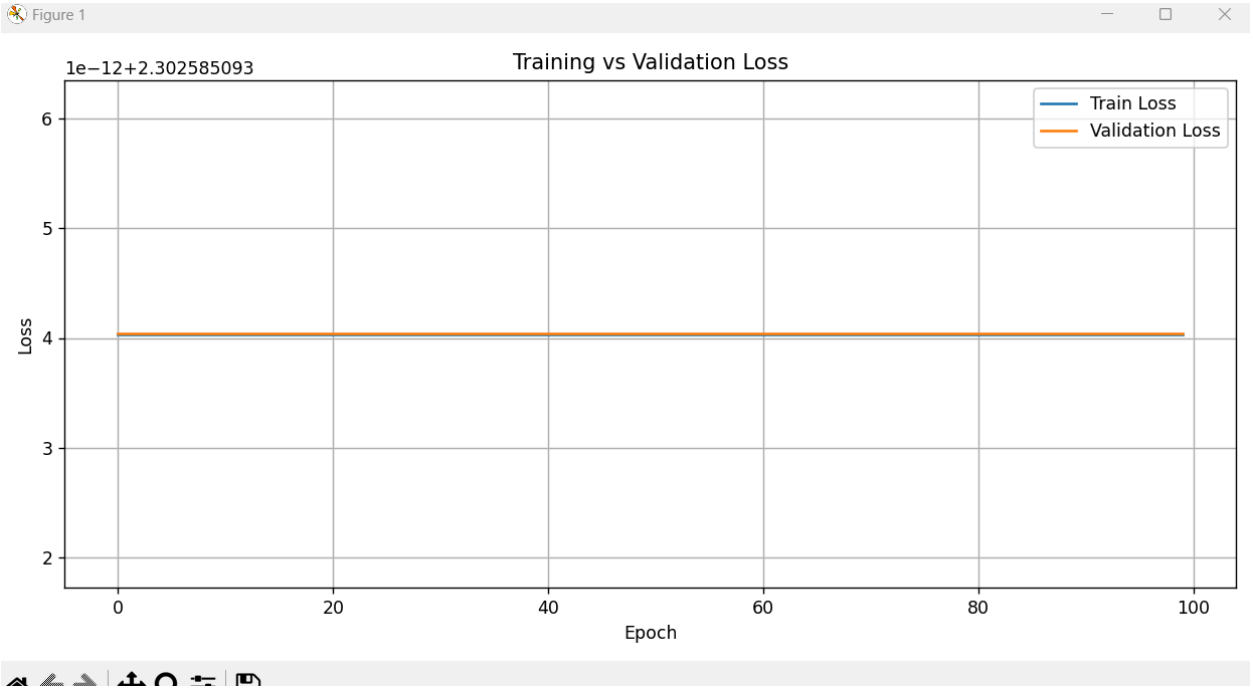


Dari ketiga variasi learning rate tersebut dapat dilihat bahwa semakin kecil learning rate maka akan menghasilkan akurasi yang semakin baik ketika epoch maksimal semakin besar. Semakin besar learning rate maka akan menghasilkan nilai konvergen yang lebih cepat tetapi berkemungkinan untuk tidak mencapai optimal global.

2.2.4. Pengaruh inisialisasi bobot

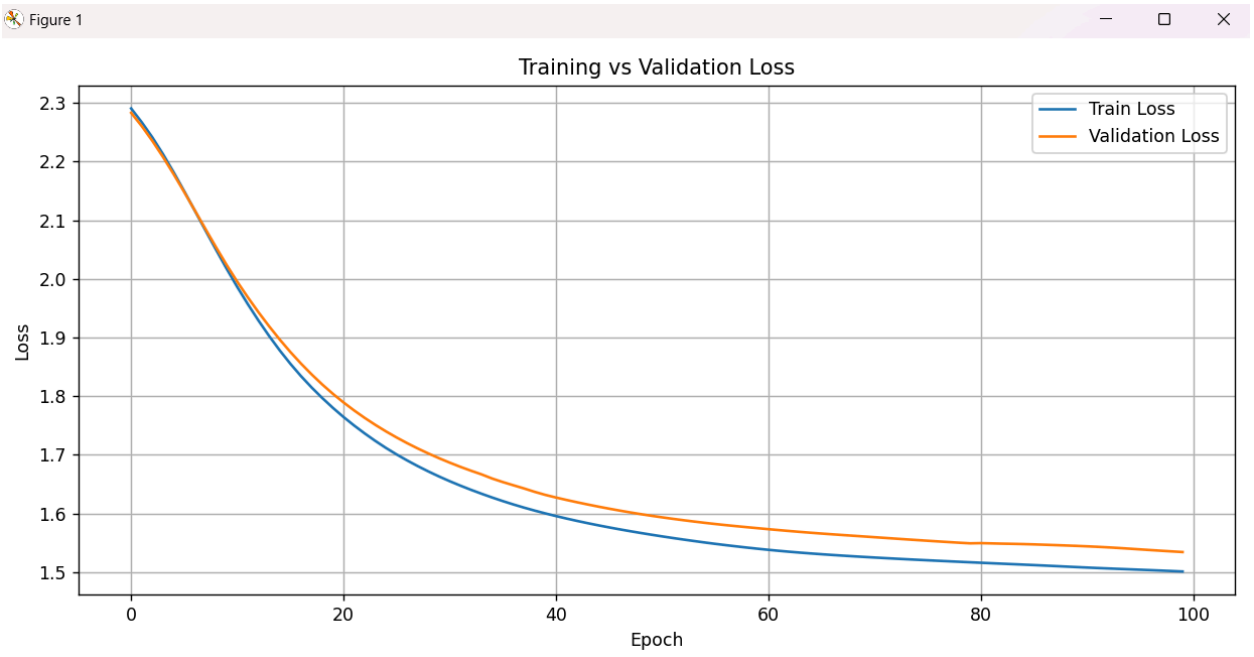
a. Zero initiation

```
Train accuracy: 0.1027
Test accuracy: 0.0973
```

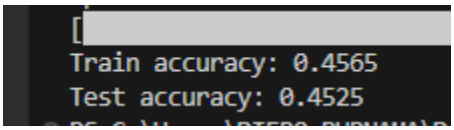


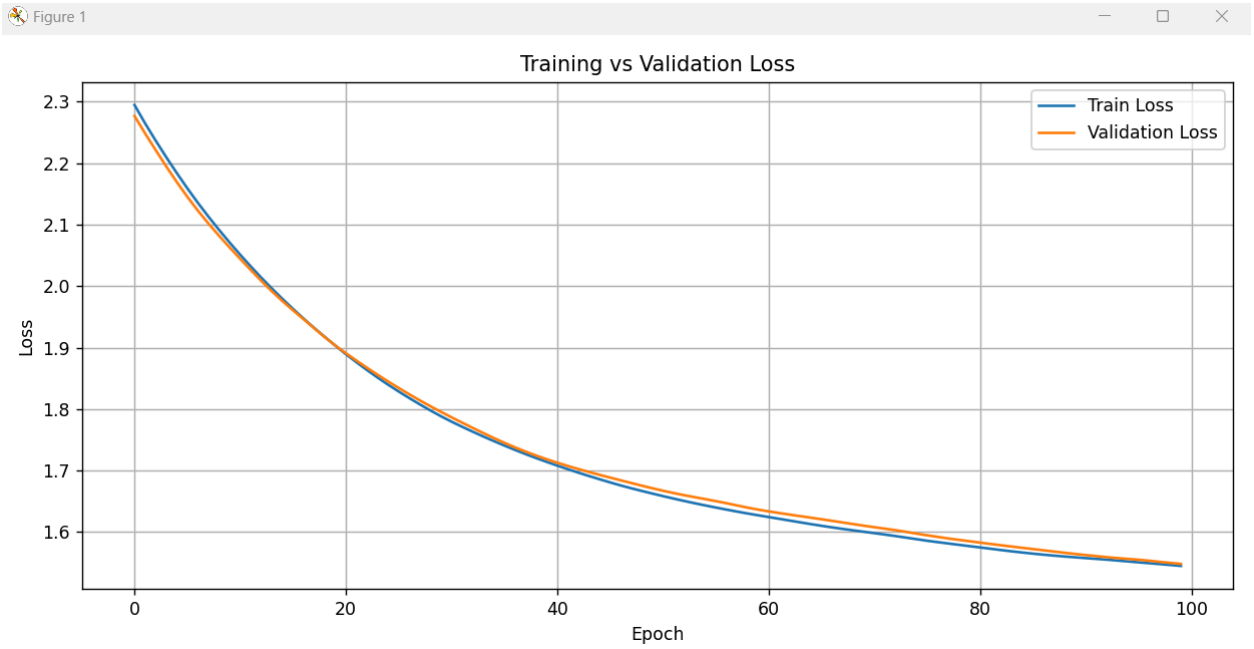
b. Uniform initiation

```
Train accuracy: 0.5135
Test accuracy: 0.4923
```



c. Normal initiation (mean = 0, variance = 0.1)





Perbedaan inisialisasi bobot menyebabkan perbedaan distribusi bobot setelah dilakukan pelatihan model. Hal ini dapat dilihat pada distribusi bobotnya yang berbeda-beda.

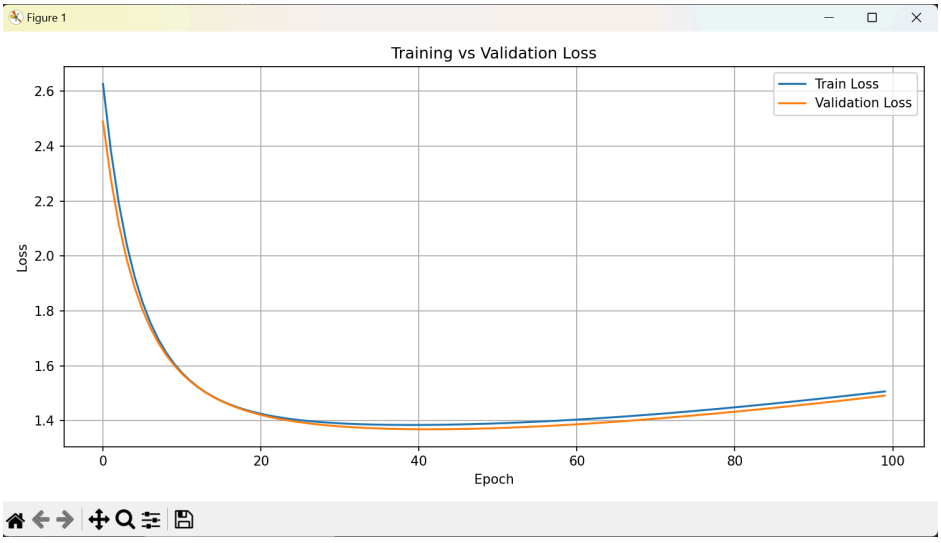
2.2.5. Pengaruh Regularisasi

d. Tanpa Regularization

```
Train accuracy: 0.8720
Test accuracy: 0.8539
```

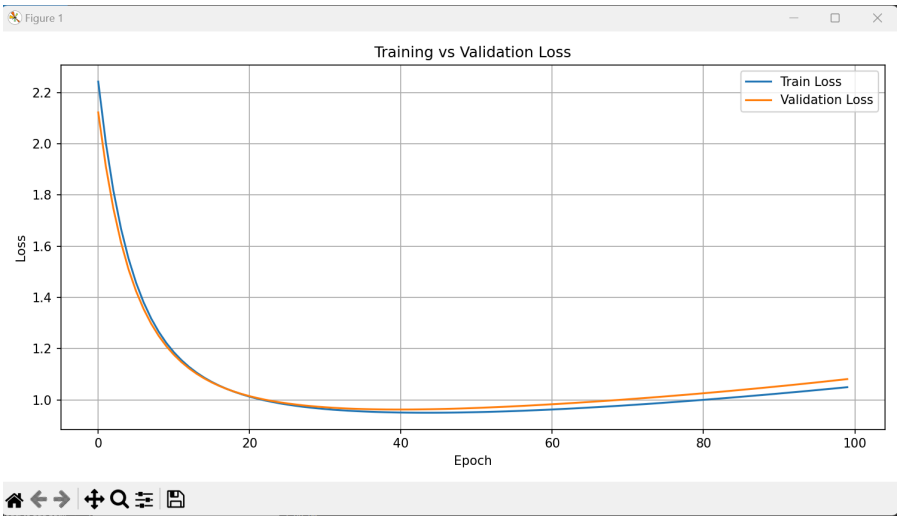
e. Dengan Regularization L1

```
Train accuracy: 0.8802
Test accuracy: 0.8581
```



f. Dengan Regularization L2

```
Train accuracy: 0.8735
Test accuracy: 0.8437
```



Regularisasi membantu model untuk mengurangi kemungkinan overfitting dengan cara memberi nilai penalti. Kedua jenis regularisasi memiliki kegunaan yang berbeda. Regularisasi L1 cenderung menghasilkan bobot yang **sparse** (banyak bobot menjadi 0) sehingga cocok untuk feature selection. Sedangkan Regularisasi L2 cenderung membuat bobot kecil (tapi jarang nol) sehingga dapat mengurangi overfitting tanpa menghilangkan fitur sepenuhnya. Perbedaan hasilnya dapat dilihat pada ketiga grafik diatas.

2.2.6. Perbandingan dengan library sklearn

Berikut adalah hasil dari perbandingan dari model FFNN buatan kelompok kamu dengan Model dari sklearn.

```
Model Train Accuracy: 0.8452
Model Test Accuracy: 0.8110
Scikit-learn Train Accuracy: 0.0986
Scikit-learn Test Accuracy: 0.0990
```

BAB III
KESIMPULAN DAN SARAN

3.1. Kesimpulan

Model FFNN adalah salah satu model dalam machine learning yang digunakan untuk memprediksi nilai atau mengambil keputusan. Terdapat banyak sekali faktor yang mempengaruhi tingkat keakuratan model tersebut. Beberapa faktor yang diuji pada tugas besar ini adalah tingkat kedalaman dan lebar, fungsi aktivasi, learning rate, dan metode inisialisasi bobot. Faktor-faktor tersebut dapat menyebabkan perbedaan yang signifikan dalam tingkat keakuratan model. Oleh karena itu, dibutuhkan pemahaman mendalam mengenai data dan model untuk mengatur parameter model sehingga mendapatkan hasil yang optimal.

3.2. Saran

1. Sebaiknya lebih mempelajari model FFNN lebih mendalam untuk mempermudah pengerjaan tugas ini.
2. Sebaiknya dapat memanajemen waktu dengan lebih baik terutama dalam pengerjaan tugas ini.

PEMBAGIAN TUGAS

Identitas Anggota	Pembagian Tugas
Rici Trisna Putra (13522026)	Backward Propagation, Fungsi Loss, Fungsi Aktivasi, Save & Load, main.py
Imam Hanif Mulyarahman (13522030)	Forward Propagation, Regularisasi, Laporan
Diero Arga Purnama (13522056)	Fungsi Aktivasi, Fungsi Loss, Kerangka Model (Neuron, Layer, FFNN), README, Visualisasi

REFERENSI

PPT ANN - FFNN IF3270 Pembelajaran Mesin
Pengenalan Deep Learning Part 1 : Neural Network
Pengenalan Deep Learning Part 3 : BackPropagation Algorithm

Repository Github