

IF702 - REDES NEURAIS

Grupo:

- Adriano Felipe Cabral Filho (afcf)
- Felipe Rodrigues de Souza (frs3)
- Lucas Vinícius José da Silva (lvjs)
- Pedro José Carneiro de Souza (pjcs)
- Vitor Sousa Silva (vss2)

Reconhecimento de Dígitos com MLPs

Importação de datasets e definição de classes

```
In [ ]: import tensorflow as tf
import keras
from keras.layers import LSTM
import numpy as np
from sklearn.metrics import accuracy_score, plot_confusion_matrix, confusion_matrix, classification_report
!pip install ipython-autotime --quiet
!pip install optuna --quiet

%load_ext autotime

| 308 kB 12.9 MB/s
| 80 kB 7.5 MB/s
| 209 kB 74.4 MB/s
| 75 kB 3.9 MB/s
| 111 kB 73.3 MB/s
| 144 kB 66.0 MB/s
| 49 kB 5.3 MB/s
Building wheel for pyperclip (setup.py) ... done
time: 264 µs (started: 2021-10-20 01:24:40 +00:00)
```

```
In [ ]: # Dataset de amostras de dígitos
mnist = tf.keras.datasets.mnist
```

time: 216 ms (started: 2021-10-20 01:24:41 +00:00)

```
In [ ]: # Category
classes = ["0", "1", "2", "3", "4", "5", "6", "7", "8", "9"]
```

time: 1.36 ms (started: 2021-10-20 01:24:42 +00:00)

Experimento

Especificação:

Para esse experimento, fixei alguns parâmetros para focar na mudança de outros Parâmetros fixados:

- Taxa de aprendizado = 0.001 (Adam Algorithm - Kingma et al., 2014)
- Número de épocas = 30

Parâmetros experimentados

- Número de camadas ocultas = 1, 2 e 4
- Número de neurônios = 16, 200
- Função de ativação = ReLu, sigmoid

```
In [ ]: #seed
tf.random.set_seed(42)
np.random.seed(42)
```

time: 1.69 ms (started: 2021-10-20 01:24:48 +00:00)

```
In [ ]: def mlp1(params):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
    model = keras.models.Sequential()
    #normalizando valores dos pixels para ser entre 0 e 1
    x_train, x_test = x_train/255.0, x_test/255.0
    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(10, activation="softmax"))
    model.compile(loss="sparse_categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
    history = model.fit(x_train, y_train, epochs=30, verbose=0)
    y_pred = np.argmax(model.predict(x_test), axis=-1)
    y_true = y_test
    print(classification_report(y_true, y_pred))

def mlp2(params):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
    model = keras.models.Sequential()
    #normalizando valores dos pixels para ser entre 0 e 1
    x_train, x_test = x_train/255.0, x_test/255.0
    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(10, activation="softmax"))
    model.compile(loss="sparse_categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
    history = model.fit(x_train, y_train, epochs=30, verbose=0)
    y_pred = np.argmax(model.predict(x_test), axis=-1)
    y_true = y_test
    print(classification_report(y_true, y_pred))

def mlp4(params):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
    model = keras.models.Sequential()
    #normalizando valores dos pixels para ser entre 0 e 1
    x_train, x_test = x_train/255.0, x_test/255.0
    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(params['neurons'], activation=params['activation_func']))
    model.add(keras.layers.Dense(10, activation="softmax"))
    model.compile(loss="sparse_categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
    history = model.fit(x_train, y_train, epochs=30, verbose=0)
    y_pred = np.argmax(model.predict(x_test), axis=-1)
    y_true = y_test
    print(classification_report(y_true, y_pred))
```

```
In [ ]: #1 camada, 16 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 16
}
mlp1(params)
```

	precision	recall	f1-score	support
0	0.97	0.98	0.97	980
1	0.98	0.98	0.98	1135
2	0.96	0.94	0.95	1032
3	0.92	0.95	0.94	1010
4	0.95	0.96	0.95	982
5	0.94	0.93	0.93	892
6	0.95	0.97	0.96	958
7	0.95	0.95	0.95	1028
8	0.94	0.94	0.94	974
9	0.95	0.92	0.93	1009
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

time: 2min 23s (started: 2021-10-17 05:44:06 +00:00)

```
In [ ]: #2 camadas, 16 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 16
}
mlp2(params)
```

	precision	recall	f1-score	support
0	0.97	0.98	0.97	980
1	0.97	0.98	0.98	1135
2	0.95	0.92	0.94	1032
3	0.91	0.94	0.93	1010
4	0.95	0.96	0.95	982
5	0.93	0.91	0.92	892
6	0.96	0.96	0.96	958
7	0.96	0.94	0.95	1028
8	0.92	0.94	0.93	974
9	0.96	0.92	0.94	1009
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

time: 2min 23s (started: 2021-10-17 05:46:30 +00:00)

```
In [ ]: #4 camadas, 16 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 16
}
mlp4(params)
```

	precision	recall	f1-score	support
0	0.96	0.98	0.97	980
1	0.98	0.97	0.98	1135
2	0.94	0.93	0.93	1032
3	0.90	0.95	0.92	1010
4	0.96	0.94	0.95	982
5	0.90	0.90	0.90	892
6	0.96	0.95	0.95	958
7	0.94	0.96	0.95	1028
8	0.93	0.91	0.92	974
9	0.95	0.92	0.94	1009
accuracy			0.94	10000
macro avg	0.94	0.94	0.94	10000
weighted avg	0.94	0.94	0.94	10000

time: 1min 49s (started: 2021-10-17 05:48:53 +00:00)

```
In [ ]: #1 camada, 200 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 200
}
mlp1(params)
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.99	0.99	0.99	1135
2	0.98	0.97	0.98	1032
3	0.97	0.99	0.98	1010
4	0.98	0.99	0.98	982
5	0.98	0.97	0.98	892
6	0.99	0.98	0.98	958
7	0.98	0.98	0.98	1028
8	0.97	0.97	0.97	974
9	0.98	0.97	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 1min 28s (started: 2021-10-17 05:50:43 +00:00)

```
In [ ]: #2 camadas, 200 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 200
}
mlp2(params)
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	980
1	0.99	0.99	0.99	1135
2	0.97	0.98	0.98	1032
3	0.96	0.98	0.97	1010
4	0.99	0.98	0.98	982
5	0.98	0.96	0.97	892

6	0.98	0.98	0.98	958
7	0.98	0.98	0.98	1028
8	0.98	0.97	0.97	974
9	0.98	0.98	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 2min 23s (started: 2021-10-17 05:52:11 +00:00)

In []:

```
#4 camadas, 200 neuronios, Sigmoid

params = {
    "activation_func": "sigmoid",
    "neurons": 200
}
mlp4(params)
```

	precision	recall	f1-score	support
0	0.98	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.98	0.97	0.98	1032
3	0.97	0.98	0.98	1010
4	0.98	0.98	0.98	982
5	0.97	0.98	0.98	892
6	0.99	0.98	0.98	958
7	0.98	0.97	0.98	1028
8	0.97	0.97	0.97	974
9	0.98	0.97	0.97	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 1min 53s (started: 2021-10-17 05:54:35 +00:00)

In []:

```
#1 camada, 16 neuronios, Relu

params = {
    "activation_func": "relu",
    "neurons": 16
}
mlp1(params)
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	980
1	0.98	0.98	0.98	1135
2	0.96	0.94	0.95	1032
3	0.93	0.95	0.94	1010
4	0.94	0.96	0.95	982
5	0.95	0.92	0.94	892
6	0.96	0.95	0.96	958
7	0.94	0.96	0.95	1028
8	0.92	0.93	0.93	974
9	0.96	0.93	0.95	1009
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

time: 1min 30s (started: 2021-10-17 05:56:28 +00:00)

In []:

```
#2 camadas, 16 neuronios, ReLu

params = {
    "activation_func": "relu",
    "neurons": 16
}
mlp2(params)
```

	precision	recall	f1-score	support
0	0.98	0.98	0.98	980
1	0.98	0.98	0.98	1135
2	0.98	0.93	0.95	1032
3	0.93	0.96	0.95	1010
4	0.93	0.97	0.95	982
5	0.94	0.93	0.93	892
6	0.98	0.94	0.96	958
7	0.94	0.97	0.96	1028
8	0.93	0.92	0.93	974
9	0.94	0.94	0.94	1009
accuracy			0.95	10000
macro avg	0.95	0.95	0.95	10000
weighted avg	0.95	0.95	0.95	10000

time: 2min 23s (started: 2021-10-17 05:57:59 +00:00)

In []:

```
#4 camadas, 16 neuronios, ReLu

params = {
    "activation_func": "relu",
    "neurons": 16
}
mlp4(params)
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	980
1	0.98	0.99	0.98	1135
2	0.98	0.92	0.95	1032
3	0.95	0.97	0.96	1010
4	0.96	0.95	0.96	982
5	0.96	0.94	0.95	892
6	0.96	0.96	0.96	958
7	0.95	0.96	0.95	1028
8	0.93	0.96	0.94	974
9	0.95	0.95	0.95	1009
accuracy			0.96	10000
macro avg	0.96	0.96	0.96	10000
weighted avg	0.96	0.96	0.96	10000

time: 1min 50s (started: 2021-10-17 06:00:23 +00:00)

In []:

```
#1 camada, 200 neuronios, Relu

params = {
    "activation_func": "relu",
    "neurons": 200
}
```

```
}
mlp1(params)
```

	precision	recall	f1-score	support
0	0.98	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.98	0.98	0.98	1032
3	0.97	0.99	0.98	1010
4	0.97	0.98	0.98	982
5	0.98	0.97	0.97	892
6	0.98	0.98	0.98	958
7	0.98	0.98	0.98	1028
8	0.98	0.97	0.98	974
9	0.98	0.97	0.98	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 2min 23s (started: 2021-10-17 06:02:13 +00:00)

```
In [ ]: #2 camadas, 200 neuronios, ReLu
```

```
params = {
    "activation_func": "relu",
    "neurons": 200
}
mlp2(params)
```

	precision	recall	f1-score	support
0	0.98	0.99	0.99	980
1	0.99	0.99	0.99	1135
2	0.99	0.97	0.98	1032
3	0.97	0.98	0.97	1010
4	0.97	0.98	0.98	982
5	0.98	0.98	0.98	892
6	0.98	0.99	0.98	958
7	0.98	0.98	0.98	1028
8	0.99	0.96	0.97	974
9	0.97	0.97	0.97	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 2min 23s (started: 2021-10-17 06:04:36 +00:00)

```
In [ ]: #4 camadas, 200 neuronios, ReLu
```

```
params = {
    "activation_func": "relu",
    "neurons": 200
}
mlp4(params)
```

	precision	recall	f1-score	support
0	1.00	0.98	0.99	980
1	0.99	0.99	0.99	1135
2	0.98	0.99	0.98	1032
3	0.98	0.98	0.98	1010
4	0.98	0.97	0.98	982
5	0.98	0.99	0.98	892
6	0.98	0.99	0.98	958
7	0.97	0.99	0.98	1028
8	0.97	0.98	0.98	974
9	0.98	0.97	0.97	1009
accuracy			0.98	10000
macro avg	0.98	0.98	0.98	10000
weighted avg	0.98	0.98	0.98	10000

time: 1min 51s (started: 2021-10-17 06:07:00 +00:00)

Sobre os resultados

Analisando os experimentos, vemos que é altamente eficaz aumentar o número de neurônios para aumentar a acurácia. Também podemos perceber que a função de ativação das camadas ocultas não têm muita influência nesse caso quando temos grande quantidade de neurônios.

Experimento com tuning

```
###Fazer aqui a especificação do experimento(verificar como foi feito no tópico "experimento")
```

```
In [ ]: import optuna
```

time: 1.7 s (started: 2021-10-20 01:26:06 +00:00)

Estudos

- Funções de ativação
 - 9 utilizadas: Sigmoid, softmax, tanh, relu, softplus, softsign, selu, elu, exponential
- Número de neurônios: valor entre 16 e 256
- Função de cálculo de perda: sparse_categorical_crossentropy
- Otimizadores: SGD, Adam, RMSprop, Adamax

```
In [ ]: # BLOCOS DE TUNING
def tuned_mlp1(trial):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()

    #setando conjunto de hiper parametros que vão ser tunados
    activation = trial.suggest_categorical('activation', ['sigmoid', 'softmax', 'tanh', 'relu', 'softplus', 'softsign', 'selu', 'elu', 'exponential'])
    n_neurons = trial.suggest_int('n_neurons', 16, 256)
    loss = trial.suggest_categorical('loss', ['sparse_categorical_crossentropy'])
    optimizer = trial.suggest_categorical('optimizer', ['sgd', 'adam', 'rmsprop', 'adamax'])
    print(activation, n_neurons, loss, optimizer)
    model = keras.models.Sequential()
    #normalizando valores dos pixels para ser entre 0 e 1
    x_train, x_test = x_train/255.0, x_test/255.0
    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(keras.layers.Dense(n_neurons, activation=activation))
    model.add(keras.layers.Dense(10, activation="softmax"))
    model.compile(loss=loss, optimizer=optimizer, metrics=["accuracy"])
    history = model.fit(x_train, y_train, epochs=30, verbose=0)
    y_pred = np.argmax(model.predict(x_test), axis=-1)
    y_true = y_test
    return accuracy_score(y_true, y_pred)

def tuned_mlp2(trial):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()
```

```

#setando conjunto de hiper parametros que vão ser tunados
activation = trial.suggest_categorical('activation', ['sigmoid', 'softmax', 'tanh', 'relu','softplus','softsign','selu','elu','exponential'])
n_neurons = trial.suggest_int('n_neurons', 16,256)
loss = trial.suggest_categorical('loss', ['sparse_categorical_crossentropy'])
optimizer = trial.suggest_categorical('optimizer', ['sgd', 'adam','rmsprop','adamax'])
print(activation,n_neurons,loss,optimizer)

model = keras.models.Sequential()
#normalizando valores dos pixels para ser entre 0 e 1
x_train, x_test = x_train/255.0, x_test/255.0
model.add(keras.layers.Flatten(input_shape=[28, 28]))
model.add(keras.layers.Dense(n_neurons, activation=activation))
model.add(keras.layers.Dense(n_neurons, activation=activation))
model.add(keras.layers.Dense(10, activation="softmax"))
model.compile(loss=loss,optimizer=optimizer,metrics=["accuracy"])
history = model.fit(x_train, y_train, epochs=30, verbose=0)
y_pred = np.argmax(model.predict(x_test), axis=-1)
y_true = y_test
return accuracy_score(y_true, y_pred)

def tuned_mlp4(trial):
    (x_train, y_train), (x_test, y_test) = mnist.load_data()

    #setando conjunto de hiper parametros que vão ser tunados
    activation = trial.suggest_categorical('activation', ['sigmoid', 'softmax', 'tanh', 'relu','softplus','softsign','selu','elu','exponential'])
    n_neurons = trial.suggest_int('n_neurons', 16,256)
    loss = trial.suggest_categorical('loss', ['sparse_categorical_crossentropy'])
    optimizer = trial.suggest_categorical('optimizer', ['sgd', 'adam','rmsprop','adamax'])
    print(activation,n_neurons,loss,optimizer)

    model = keras.models.Sequential()
    #normalizando valores dos pixels para ser entre 0 e 1
    x_train, x_test = x_train/255.0, x_test/255.0
    model.add(keras.layers.Flatten(input_shape=[28, 28]))
    model.add(keras.layers.Dense(n_neurons, activation=activation))
    model.add(keras.layers.Dense(n_neurons, activation=activation))
    model.add(keras.layers.Dense(n_neurons, activation=activation))
    model.add(keras.layers.Dense(n_neurons, activation=activation))
    model.add(keras.layers.Dense(10, activation="softmax"))
    model.compile(loss=loss,optimizer=optimizer,metrics=["accuracy"])
    history = model.fit(x_train, y_train, epochs=30, verbose=0)
    y_pred = np.argmax(model.predict(x_test), axis=-1)
    y_true = y_test
    return accuracy_score(y_true, y_pred)

```

time: 101 ms (started: 2021-10-20 04:41:22 +00:00)

In []:

```

study_1 = optuna.create_study(direction='maximize')
study_1.optimize(tuned_mlp1,n_trials=32)

[I 2021-10-20 02:19:49,641] A new study created in memory with name: no-name-8448f802-55a0-4afe-bc55-aff4c98f7a99
selu 22 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:21:12,816] Trial 0 finished with value: 0.9564 and parameters: {'activation': 'selu', 'n_neurons': 22, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 0 with value: 0.9564.
softplus 18 sparse_categorical_crossentropy adam
[I 2021-10-20 02:22:35,889] Trial 1 finished with value: 0.9589 and parameters: {'activation': 'softplus', 'n_neurons': 18, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 1 with value: 0.9589.
relu 215 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:24:14,338] Trial 2 finished with value: 0.9751 and parameters: {'activation': 'relu', 'n_neurons': 215, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9751.
softmax 111 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:25:34,970] Trial 3 finished with value: 0.9631 and parameters: {'activation': 'softmax', 'n_neurons': 111, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9751.
softmax 192 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 02:28:02,231] Trial 4 finished with value: 0.941 and parameters: {'activation': 'softmax', 'n_neurons': 192, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 2 with value: 0.9751.
relu 31 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:28:59,886] Trial 5 finished with value: 0.9591 and parameters: {'activation': 'relu', 'n_neurons': 31, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9751.
softmax 130 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 02:31:18,100] Trial 6 finished with value: 0.9366 and parameters: {'activation': 'softmax', 'n_neurons': 130, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 2 with value: 0.9751.
exponential 214 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:32:54,418] Trial 7 finished with value: 0.9738 and parameters: {'activation': 'exponential', 'n_neurons': 214, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9751.
selu 130 sparse_categorical_crossentropy sgd
[I 2021-10-20 02:34:22,808] Trial 8 finished with value: 0.9661 and parameters: {'activation': 'selu', 'n_neurons': 130, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9751.
relu 125 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 02:36:35,691] Trial 9 finished with value: 0.9745 and parameters: {'activation': 'relu', 'n_neurons': 125, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 2 with value: 0.9751.
softsign 250 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:38:32,846] Trial 10 finished with value: 0.9801 and parameters: {'activation': 'softsign', 'n_neurons': 250, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 10 with value: 0.9801.
softsign 256 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:40:27,548] Trial 11 finished with value: 0.9814 and parameters: {'activation': 'softsign', 'n_neurons': 256, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
softsign 245 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:42:28,311] Trial 12 finished with value: 0.9801 and parameters: {'activation': 'softsign', 'n_neurons': 245, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
softsign 251 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:44:36,325] Trial 13 finished with value: 0.9805 and parameters: {'activation': 'softsign', 'n_neurons': 251, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
softsign 173 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:47:00,020] Trial 14 finished with value: 0.9779 and parameters: {'activation': 'softsign', 'n_neurons': 173, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
tanh 80 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:48:19,783] Trial 15 finished with value: 0.9768 and parameters: {'activation': 'tanh', 'n_neurons': 80, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
elu 172 sparse_categorical_crossentropy adamax
[I 2021-10-20 02:50:09,400] Trial 16 finished with value: 0.9795 and parameters: {'activation': 'elu', 'n_neurons': 172, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9814.
sigmoid 255 sparse_categorical_crossentropy adam
[I 2021-10-20 02:52:08,207] Trial 17 finished with value: 0.9815 and parameters: {'activation': 'sigmoid', 'n_neurons': 255, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9815.
sigmoid 222 sparse_categorical_crossentropy adam
[I 2021-10-20 02:53:59,783] Trial 18 finished with value: 0.9811 and parameters: {'activation': 'sigmoid', 'n_neurons': 222, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9815.
sigmoid 164 sparse_categorical_crossentropy adam
[I 2021-10-20 02:55:40,417] Trial 19 finished with value: 0.9789 and parameters: {'activation': 'sigmoid', 'n_neurons': 164, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9815.
sigmoid 228 sparse_categorical_crossentropy adam
[I 2021-10-20 02:57:33,313] Trial 20 finished with value: 0.9816 and parameters: {'activation': 'sigmoid', 'n_neurons': 228, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
sigmoid 227 sparse_categorical_crossentropy adam
[I 2021-10-20 02:59:57,018] Trial 21 finished with value: 0.9806 and parameters: {'activation': 'sigmoid', 'n_neurons': 227, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.

```

```
sigmoid 194 sparse_categorical_crossentropy adam
[I 2021-10-20 03:01:44,059] Trial 22 finished with value: 0.9798 and parameters: {'activation': 'sigmoid', 'n_neurons': 194, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
sigmoid 235 sparse_categorical_crossentropy adam
[I 2021-10-20 03:04:08,113] Trial 23 finished with value: 0.9806 and parameters: {'activation': 'sigmoid', 'n_neurons': 235, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
elu 255 sparse_categorical_crossentropy adam
[I 2021-10-20 03:06:02,586] Trial 24 finished with value: 0.9754 and parameters: {'activation': 'elu', 'n_neurons': 255, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
exponential 201 sparse_categorical_crossentropy adam
[I 2021-10-20 03:07:48,528] Trial 25 finished with value: 0.098 and parameters: {'activation': 'exponential', 'n_neurons': 201, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
softplus 155 sparse_categorical_crossentropy adam
[I 2021-10-20 03:10:11,686] Trial 26 finished with value: 0.9816 and parameters: {'activation': 'softplus', 'n_neurons': 155, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
softplus 156 sparse_categorical_crossentropy adam
[I 2021-10-20 03:12:34,857] Trial 27 finished with value: 0.98 and parameters: {'activation': 'softplus', 'n_neurons': 156, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
softplus 59 sparse_categorical_crossentropy adam
[I 2021-10-20 03:13:57,989] Trial 28 finished with value: 0.9761 and parameters: {'activation': 'softplus', 'n_neurons': 59, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
selu 90 sparse_categorical_crossentropy adam
[I 2021-10-20 03:15:32,083] Trial 29 finished with value: 0.9757 and parameters: {'activation': 'selu', 'n_neurons': 90, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
tanh 185 sparse_categorical_crossentropy adam
[I 2021-10-20 03:17:14,109] Trial 30 finished with value: 0.9814 and parameters: {'activation': 'tanh', 'n_neurons': 185, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
tanh 151 sparse_categorical_crossentropy adam
[I 2021-10-20 03:19:37,250] Trial 31 finished with value: 0.9752 and parameters: {'activation': 'tanh', 'n_neurons': 151, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 20 with value: 0.9816.
time: 59min 47s (started: 2021-10-20 02:19:49 +00:00)
```

```
In [ ]: study_2 = optuna.create_study(direction='maximize')
study_2.optimize(tuned_mlp2,n_trials=32)
```

```
[I 2021-10-20 03:24:23,011] A new study created in memory with name: no-name-008273fd-c44d-40f6-aa94-e302fb6c8474
tanh 51 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 03:25:44,642] Trial 0 finished with value: 0.9702 and parameters: {'activation': 'tanh', 'n_neurons': 51, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 0 with value: 0.9702.
softmax 191 sparse_categorical_crossentropy sgd
[I 2021-10-20 03:28:07,857] Trial 1 finished with value: 0.1135 and parameters: {'activation': 'softmax', 'n_neurons': 191, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 0 with value: 0.9702.
tanh 79 sparse_categorical_crossentropy sgd
[I 2021-10-20 03:29:22,036] Trial 2 finished with value: 0.9728 and parameters: {'activation': 'tanh', 'n_neurons': 79, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 2 with value: 0.9728.
relu 250 sparse_categorical_crossentropy adam
[I 2021-10-20 03:31:52,833] Trial 3 finished with value: 0.9817 and parameters: {'activation': 'relu', 'n_neurons': 250, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
relu 44 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 03:33:16,142] Trial 4 finished with value: 0.9708 and parameters: {'activation': 'relu', 'n_neurons': 44, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 3 with value: 0.9817.
softmax 220 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 03:36:39,975] Trial 5 finished with value: 0.9455 and parameters: {'activation': 'softmax', 'n_neurons': 220, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 3 with value: 0.9817.
exponential 231 sparse_categorical_crossentropy adamax
[I 2021-10-20 03:39:05,976] Trial 6 finished with value: 0.9773 and parameters: {'activation': 'exponential', 'n_neurons': 231, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 3 with value: 0.9817.
tanh 219 sparse_categorical_crossentropy adam
[I 2021-10-20 03:41:29,247] Trial 7 finished with value: 0.9768 and parameters: {'activation': 'tanh', 'n_neurons': 219, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
selu 43 sparse_categorical_crossentropy adamax
[I 2021-10-20 03:42:52,918] Trial 8 finished with value: 0.9713 and parameters: {'activation': 'selu', 'n_neurons': 43, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 3 with value: 0.9817.
softmax 48 sparse_categorical_crossentropy sgd
[I 2021-10-20 03:44:02,461] Trial 9 finished with value: 0.1135 and parameters: {'activation': 'softmax', 'n_neurons': 48, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 3 with value: 0.9817.
sigmoid 146 sparse_categorical_crossentropy adam
[I 2021-10-20 03:45:55,508] Trial 10 finished with value: 0.9796 and parameters: {'activation': 'sigmoid', 'n_neurons': 146, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
sigmoid 138 sparse_categorical_crossentropy adam
[I 2021-10-20 03:48:18,724] Trial 11 finished with value: 0.9758 and parameters: {'activation': 'sigmoid', 'n_neurons': 138, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
softplus 155 sparse_categorical_crossentropy adam
[I 2021-10-20 03:50:26,218] Trial 12 finished with value: 0.9809 and parameters: {'activation': 'softplus', 'n_neurons': 155, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
softplus 174 sparse_categorical_crossentropy adam
[I 2021-10-20 03:52:40,631] Trial 13 finished with value: 0.9801 and parameters: {'activation': 'softplus', 'n_neurons': 174, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
softsign 256 sparse_categorical_crossentropy adam
[I 2021-10-20 03:56:04,172] Trial 14 finished with value: 0.979 and parameters: {'activation': 'softsign', 'n_neurons': 256, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
softplus 110 sparse_categorical_crossentropy adam
[I 2021-10-20 03:58:27,362] Trial 15 finished with value: 0.9774 and parameters: {'activation': 'softplus', 'n_neurons': 110, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
relu 180 sparse_categorical_crossentropy adam
[I 2021-10-20 04:00:36,459] Trial 16 finished with value: 0.9805 and parameters: {'activation': 'relu', 'n_neurons': 180, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
elu 92 sparse_categorical_crossentropy adam
[I 2021-10-20 04:02:28,563] Trial 17 finished with value: 0.9771 and parameters: {'activation': 'elu', 'n_neurons': 92, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 3 with value: 0.9817.
softplus 252 sparse_categorical_crossentropy adamax
[I 2021-10-20 04:05:33,366] Trial 18 finished with value: 0.9797 and parameters: {'activation': 'softplus', 'n_neurons': 252, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 3 with value: 0.9817.
relu 158 sparse_categorical_crossentropy adam
[I 2021-10-20 04:07:56,590] Trial 19 finished with value: 0.9818 and parameters: {'activation': 'relu', 'n_neurons': 158, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 19 with value: 0.9818.
relu 105 sparse_categorical_crossentropy adam
[I 2021-10-20 04:10:19,788] Trial 20 finished with value: 0.9784 and parameters: {'activation': 'relu', 'n_neurons': 105, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 19 with value: 0.9818.
relu 159 sparse_categorical_crossentropy adam
[I 2021-10-20 04:12:43,134] Trial 21 finished with value: 0.9802 and parameters: {'activation': 'relu', 'n_neurons': 159, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 19 with value: 0.9818.
relu 203 sparse_categorical_crossentropy adam
[I 2021-10-20 04:14:56,272] Trial 22 finished with value: 0.982 and parameters: {'activation': 'relu', 'n_neurons': 203, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 22 with value: 0.982.
relu 207 sparse_categorical_crossentropy adam
[I 2021-10-20 04:17:19,538] Trial 23 finished with value: 0.9822 and parameters: {'activation': 'relu', 'n_neurons': 207, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 23 with value: 0.9822.
relu 207 sparse_categorical_crossentropy adam
[I 2021-10-20 04:19:43,570] Trial 24 finished with value: 0.9789 and parameters: {'activation': 'relu', 'n_neurons': 207, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 23 with value: 0.9822.
relu 198 sparse_categorical_crossentropy adam
[I 2021-10-20 04:21:55,156] Trial 25 finished with value: 0.9803 and parameters: {'activation': 'relu', 'n_neurons': 198, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 23 with value: 0.9822.
softsign 127 sparse_categorical_crossentropy sgd
```

```
[I 2021-10-20 04:23:37,264] Trial 26 finished with value: 0.9692 and parameters: {'activation': 'softsign', 'n_neurons': 127, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 23 with value: 0.9822.
exponential 171 sparse_categorical_crossentropy adamax
[I 2021-10-20 04:25:53,110] Trial 27 finished with value: 0.9753 and parameters: {'activation': 'exponential', 'n_neurons': 171, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 23 with value: 0.9822.
elu 194 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 04:29:16,741] Trial 28 finished with value: 0.9811 and parameters: {'activation': 'elu', 'n_neurons': 194, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 23 with value: 0.9822.
relu 229 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 04:32:32,276] Trial 29 finished with value: 0.9767 and parameters: {'activation': 'relu', 'n_neurons': 229, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 23 with value: 0.9822.
selu 125 sparse_categorical_crossentropy adam
[I 2021-10-20 04:34:55,478] Trial 30 finished with value: 0.9805 and parameters: {'activation': 'selu', 'n_neurons': 125, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 23 with value: 0.9822.
relu 238 sparse_categorical_crossentropy adam
[I 2021-10-20 04:37:15,408] Trial 31 finished with value: 0.9814 and parameters: {'activation': 'relu', 'n_neurons': 238, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 23 with value: 0.9822.
time: 1h 12min 52s (started: 2021-10-20 03:24:23 +00:00)
```

In []:

```
study_4 = optuna.create_study(direction='maximize')
study_4.optimize(tuned_mlp4,n_trials=32)
```

```
[I 2021-10-20 04:41:34,192] A new study created in memory with name: no-name-350ca4a3-1a6f-443c-a5ed-8136e4cdb700
softplus 17 sparse_categorical_crossentropy adam
[I 2021-10-20 04:42:57,438] Trial 0 finished with value: 0.9639 and parameters: {'activation': 'softplus', 'n_neurons': 17, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 0 with value: 0.9639.
tanh 209 sparse_categorical_crossentropy adamax
[I 2021-10-20 04:46:14,949] Trial 1 finished with value: 0.9806 and parameters: {'activation': 'tanh', 'n_neurons': 209, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 1 with value: 0.9806.
tanh 139 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 04:49:17,385] Trial 2 finished with value: 0.9774 and parameters: {'activation': 'tanh', 'n_neurons': 139, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 1 with value: 0.9806.
softsign 25 sparse_categorical_crossentropy adam
[I 2021-10-20 04:50:41,766] Trial 3 finished with value: 0.9643 and parameters: {'activation': 'softsign', 'n_neurons': 25, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 1 with value: 0.9806.
sigmoid 113 sparse_categorical_crossentropy sgd
[I 2021-10-20 04:52:32,059] Trial 4 finished with value: 0.8731 and parameters: {'activation': 'sigmoid', 'n_neurons': 113, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 1 with value: 0.9806.
exponential 222 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 04:56:45,930] Trial 5 finished with value: 0.098 and parameters: {'activation': 'exponential', 'n_neurons': 222, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 1 with value: 0.9806.
elu 135 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:00:10,341] Trial 6 finished with value: 0.9798 and parameters: {'activation': 'elu', 'n_neurons': 135, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 1 with value: 0.9806.
softplus 115 sparse_categorical_crossentropy sgd
[I 2021-10-20 05:02:14,881] Trial 7 finished with value: 0.9708 and parameters: {'activation': 'softplus', 'n_neurons': 115, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 1 with value: 0.9806.
relu 157 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 05:05:33,301] Trial 8 finished with value: 0.975 and parameters: {'activation': 'relu', 'n_neurons': 157, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 1 with value: 0.9806.
elu 52 sparse_categorical_crossentropy rmsprop
[I 2021-10-20 05:07:56,715] Trial 9 finished with value: 0.9687 and parameters: {'activation': 'elu', 'n_neurons': 52, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'rmsprop'}. Best is trial 1 with value: 0.9806.
softmax 243 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:11:50,085] Trial 10 finished with value: 0.1135 and parameters: {'activation': 'softmax', 'n_neurons': 243, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 1 with value: 0.9806.
elu 193 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:14:57,845] Trial 11 finished with value: 0.9816 and parameters: {'activation': 'elu', 'n_neurons': 193, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9816.
tanh 198 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:18:00,358] Trial 12 finished with value: 0.9816 and parameters: {'activation': 'tanh', 'n_neurons': 198, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9816.
selu 189 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:21:04,667] Trial 13 finished with value: 0.9816 and parameters: {'activation': 'selu', 'n_neurons': 189, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9816.
elu 180 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:24:01,762] Trial 14 finished with value: 0.9808 and parameters: {'activation': 'elu', 'n_neurons': 180, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 11 with value: 0.9816.
tanh 255 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:28:25,246] Trial 15 finished with value: 0.9824 and parameters: {'activation': 'tanh', 'n_neurons': 255, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 15 with value: 0.9824.
sigmoid 253 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:32:20,920] Trial 16 finished with value: 0.9782 and parameters: {'activation': 'sigmoid', 'n_neurons': 253, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 15 with value: 0.9824.
selu 236 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:36:44,532] Trial 17 finished with value: 0.9828 and parameters: {'activation': 'selu', 'n_neurons': 236, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
selu 224 sparse_categorical_crossentropy sgd
[I 2021-10-20 05:39:46,025] Trial 18 finished with value: 0.9785 and parameters: {'activation': 'selu', 'n_neurons': 224, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 17 with value: 0.9828.
selu 255 sparse_categorical_crossentropy adam
[I 2021-10-20 05:43:41,691] Trial 19 finished with value: 0.9781 and parameters: {'activation': 'selu', 'n_neurons': 255, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9828.
exponential 79 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:45:28,082] Trial 20 finished with value: 0.098 and parameters: {'activation': 'exponential', 'n_neurons': 79, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
selu 229 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:49:10,239] Trial 21 finished with value: 0.98 and parameters: {'activation': 'selu', 'n_neurons': 229, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
selu 174 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:52:01,437] Trial 22 finished with value: 0.9812 and parameters: {'activation': 'selu', 'n_neurons': 174, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
softsign 215 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:55:22,924] Trial 23 finished with value: 0.9801 and parameters: {'activation': 'softsign', 'n_neurons': 215, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
relu 236 sparse_categorical_crossentropy adamax
[I 2021-10-20 05:59:46,532] Trial 24 finished with value: 0.9828 and parameters: {'activation': 'relu', 'n_neurons': 236, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
relu 242 sparse_categorical_crossentropy adamax
[I 2021-10-20 06:04:10,062] Trial 25 finished with value: 0.9756 and parameters: {'activation': 'relu', 'n_neurons': 242, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
relu 238 sparse_categorical_crossentropy adamax
[I 2021-10-20 06:07:50,357] Trial 26 finished with value: 0.9788 and parameters: {'activation': 'relu', 'n_neurons': 238, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
tanh 208 sparse_categorical_crossentropy adam
[I 2021-10-20 06:10:53,165] Trial 27 finished with value: 0.9776 and parameters: {'activation': 'tanh', 'n_neurons': 208, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9828.
softmax 165 sparse_categorical_crossentropy sgd
[I 2021-10-20 06:13:25,557] Trial 28 finished with value: 0.1135 and parameters: {'activation': 'softmax', 'n_neurons': 165, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'sgd'}. Best is trial 17 with value: 0.9828.
softplus 254 sparse_categorical_crossentropy adam
[I 2021-10-20 06:17:41,691] Trial 29 finished with value: 0.9816 and parameters: {'activation': 'softplus', 'n_neurons': 254, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9828.
relu 233 sparse_categorical_crossentropy adamax
[I 2021-10-20 06:21:20,465] Trial 30 finished with value: 0.9804 and parameters: {'activation': 'relu', 'n_neurons': 233, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}. Best is trial 17 with value: 0.9828.
```



```
softplus 252 sparse_categorical_crossentropy adam
[I 2021-10-20 06:25:39,390] Trial 31 finished with value: 0.9819 and parameters: {'activation': 'softplus', 'n_neurons': 252, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}. Best is trial 17 with value: 0.9828.
time: 1h 44min 5s (started: 2021-10-20 04:41:34 +00:00)
```

```
In [ ]: trial = study_1.best_trial
print(f'Accuracy: {trial.value}')
print(f'Best hyperparameters: {trial.params}')
print(trial)
```

Accuracy: 0.9816
Best hyperparameters: {'activation': 'sigmoid', 'n_neurons': 228, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}
FrozenTrial(number=20, values=[0.9816], datetime_start=datetime.datetime(2021, 10, 20, 2, 55, 40, 419629), datetime_complete=datetime.datetime(2021, 10, 20, 2, 57, 33, 313460), params={'activation': 'sigmoid', 'n_neurons': 228, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}, distributions={'activation': CategoricalDistribution(choices=('sigmoid', 'softmax', 'tanh', 'relu', 'softplus', 'softsign', 'selu', 'elu', 'exponential')), 'n_neurons': IntUniformDistribution(high=256, low=16, step=1), 'loss': CategoricalDistribution(choices=('sparse_categorical_crossentropy',)), 'optimizer': CategoricalDistribution(choices=('sgd', 'adam', 'rmsprop', 'adamax'))}, user_attrs={}, system_attrs={}, intermediate_values={}, trial_id=20, state=TrialState.COMPLETE, value=None)
time: 4.85 ms (started: 2021-10-20 06:26:53 +00:00)

```
In [ ]: trial = study_2.best_trial
print(f'Accuracy: {trial.value}')
print(f'Best hyperparameters: {trial.params}')
print(trial)
```

Accuracy: 0.9822
Best hyperparameters: {'activation': 'relu', 'n_neurons': 207, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}
FrozenTrial(number=23, values=[0.9822], datetime_start=datetime.datetime(2021, 10, 20, 4, 14, 56, 274775), datetime_complete=datetime.datetime(2021, 10, 20, 4, 17, 19, 538143), params={'activation': 'relu', 'n_neurons': 207, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adam'}, distributions={'activation': CategoricalDistribution(choices=('sigmoid', 'softmax', 'tanh', 'relu', 'softplus', 'softsign', 'selu', 'elu', 'exponential')), 'n_neurons': IntUniformDistribution(high=256, low=16, step=1), 'loss': CategoricalDistribution(choices=('sparse_categorical_crossentropy',)), 'optimizer': CategoricalDistribution(choices=('sgd', 'adam', 'rmsprop', 'adamax'))}, user_attrs={}, system_attrs={}, intermediate_values={}, trial_id=23, state=TrialState.COMPLETE, value=None)
time: 2.64 ms (started: 2021-10-20 06:27:17 +00:00)

```
In [ ]: trial = study_4.best_trial
print(f'Accuracy: {trial.value}')
print(f'Best hyperparameters: {trial.params}')
print(trial)
```

Accuracy: 0.9828
Best hyperparameters: {'activation': 'selu', 'n_neurons': 236, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}
FrozenTrial(number=17, values=[0.9828], datetime_start=datetime.datetime(2021, 10, 20, 5, 32, 20, 921783), datetime_complete=datetime.datetime(2021, 10, 20, 5, 36, 44, 532400), params={'activation': 'selu', 'n_neurons': 236, 'loss': 'sparse_categorical_crossentropy', 'optimizer': 'adamax'}, distributions={'activation': CategoricalDistribution(choices=('sigmoid', 'softmax', 'tanh', 'relu', 'softplus', 'softsign', 'selu', 'elu', 'exponential')), 'n_neurons': IntUniformDistribution(high=256, low=16, step=1), 'loss': CategoricalDistribution(choices=('sparse_categorical_crossentropy',)), 'optimizer': CategoricalDistribution(choices=('sgd', 'adam', 'rmsprop', 'adamax'))}, user_attrs={}, system_attrs={}, intermediate_values={}, trial_id=17, state=TrialState.COMPLETE, value=None)
time: 2.14 ms (started: 2021-10-20 06:32:08 +00:00)

Primeiro estudo

- Acurácia: 98.16%
- Melhor configuração de hiperparâmetros:
- Função de ativação: Sigmoid
- Número de neurônios: 228
- Otimizador: Adam

Segundo estudo

- Acurácia: 98.22%
- Melhor configuração de hiperparâmetros:
- Função de ativação: Relu
- Número de neurônios: 207
- Otimizador: Adam

Terceiro estudo

- Acurácia: 98.28%
- Melhor configuração de hiperparâmetros:
- Função de ativação: Selu
- Número de neurônios: 236
- Otimizador: Adamax

Concluímos que nos três estudos obtivemos resultados bastante semelhantes, dado que a variação da acurácia entre o melhor e o pior resultado foi de apenas 0.12%. Também percebemos que a função de ativação foi distinta nos três casos, e que o otimizador Adam foi o melhor em 2 dos nossos estudos.

Resultados

```
In [ ]: optuna.visualization.plot_optimization_history(study_1)
```

time: 1.34 s (started: 2021-10-20 06:27:46 +00:00)

Estudo 1

Houve alguns outliers no testes 3, mas no geral, todos os outros testes atingiram o melhor valor possível.




```
In [ ]: optuna.visualization.plot_optimization_history(study_2)
```

time: 176 ms (started: 2021-10-20 06:28:04 +00:00)

Estudo 2

Houveram apenas 2 outliers distoantes, todos os outros valores foram consistentes em atingir o melhor valor possível.

```
In [ ]: optuna.visualization.plot_optimization_history(study_4)
```

time: 187 ms (started: 2021-10-20 06:28:43 +00:00)

Estudo 3

Este estudo teve 5 pontos como outliers, sendo este o pior de nossos cenários.

```
In [ ]: optuna.visualization.plot_slice(study_1)
```

time: 525 ms (started: 2021-10-20 06:29:11 +00:00)

Avaliação dos parâmetros do estudo 1

- Para as funções de ativação, todas elas demonstraram estar no mesmo espectro, houve um outlier na exponencial e na softmax.
 - Softsign, tan, elu e sigmoid bastaram ser bastante consistentes em suas pontuações.

- Selu, softplus, relu e softmax foram mais dispersas.
- Para quantidade de neurônios houveram apenas 2 outliers. Os valores mais a esquerda (com menos neurônios) se provaram um pouco menos eficientes que os mais a direita (com mais neurônios).
- Quanto ao otimizador, houveram outliers na SGD e no Adam, sendo o Adamax a mais consistente em apresentar bons resultados.

```
In [ ]: optuna.visualization.plot_slice(study_2)
```

time: 283 ms (started: 2021-10-20 06:29:19 +00:00)

Avaliação dos parâmetros do estudo 2

- Mais uma vez, a função de ativação softmax se provou um outlier, enquanto as outras mantiveram bons resultados.
- Houveram 2 outliers no número de neurônios, e, desta vez, não é tão perceptível a evolução com aumento do número de neurônios vista no estudo 1.
- SGD foi o único outlier dentre os otimizadores, os outros apresentaram consistência em bons resultados.

```
In [ ]: optuna.visualization.plot_slice(study_4)
```

time: 298 ms (started: 2021-10-20 06:29:28 +00:00)

Avaliação dos parâmetros do estudo 3

- A função de ativação softmax (como vista nos dois estudos anteriores) continua sendo um outlier, e, de forma semelhante ao que havia acontecido no estudo 1, a função exponencial também foi um outlier, a sigmoid apresentou compartimento um pouco abaixo em um de seus testes. No geral, as outras funções foram consistentes em obter bons resultados.
- Surpreendentemente houveram mais outliers quando foram adicionados mais neurônios neste estudo.
- SGD, Adamax e RMSprop foram os outlier dentre os otimizadores, apenas a Adam se mostrou consistente nos seus resultados.

Importância dos parâmetros

Para todos os nossos estudos, a função de ativação foi o parâmetro que mais importou (sempre com mais de 67% de importância, ou seja mais de 2/3), assim como o otimizador ficou em segundo lugar em todos e o número de neurônios em terceiro.

```
In [ ]: optuna.visualization.plot_param_importances(study_1)
```

time: 1.11 s (started: 2021-10-20 06:30:07 +00:00)

```
In [ ]: optuna.visualization.plot_param_importances(study_2)
```

time: 954 ms (started: 2021-10-20 06:30:18 +00:00)

```
In [ ]: optuna.visualization.plot_param_importances(study_4)
```

time: 1.14 s (started: 2021-10-20 06:30:28 +00:00)

Sobre os resultados

###Fazer aqui análise comparativa do tuning aqui

Reconhecimento de Dígitos com CNN