Shri Ramdeobaba College of Engineering and Management
Nagpur, 440013

Department of Computer Science Engineering (AIML)

Deep Learning Lab

Name : *Shantanu Mane*
Roll No. : *E63*
Batch : *CSE-AIML*
Date : *5/3/2023*

AIM - To implement and evaluate the performance of the following neural network architectures with 2 hidden layers on the SONAR dataset for 200 epochs.

1. Backpropagation NN with adam optimizer.
2. l1 and l2 regularizations.
3. early stopping with p = 5
4. dropout regularization with p = 0.2 for input and second hidden layer.

Importing the Dependancies

```
import tensorflow as tf
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
```

Loading Our Data

```
# Load the Sonar dataset
X, y = make_classification(n_samples=208, n_features=60, n_informative=60, n_redundant=0,

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Creating a Sequential Model

```
%%time

nn = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32, input_dim=60, activation='relu'),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])


nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

nn.fit(X_train, y_train, epochs=200)
```

```
Epoch 1/200
6/6 [==============================] - 1s 2ms/step - loss: 1.7980 - accuracy: 0.5060
Epoch 2/200
6/6 [==============================] - 0s 2ms/step - loss: 1.2402 - accuracy: 0.5361
Epoch 3/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9177 - accuracy: 0.5843
Epoch 4/200
6/6 [==============================] - 0s 2ms/step - loss: 0.7141 - accuracy: 0.6205
Epoch 5/200
6/6 [==============================] - 0s 2ms/step - loss: 0.6007 - accuracy: 0.6747
Epoch 6/200
6/6 [==============================] - 0s 3ms/step - loss: 0.5450 - accuracy: 0.7289
Epoch 7/200
6/6 [==============================] - 0s 2ms/step - loss: 0.4978 - accuracy: 0.7711
Epoch 8/200
6/6 [==============================] - 0s 2ms/step - loss: 0.4561 - accuracy: 0.7952
Epoch 9/200
6/6 [==============================] - 0s 2ms/step - loss: 0.4215 - accuracy: 0.8133
Epoch 10/200
6/6 [==============================] - 0s 2ms/step - loss: 0.3902 - accuracy: 0.8373

<keras.callbacks.History at 0x7f8263244970>
```

```
%%time

nn_with_reg = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32, input_dim=60, activation='relu', kernel_regularizer=tf.keras
    tf.keras.layers.Dense(16, activation='relu',kernel_regularizer=tf.keras.regularizers.l
    tf.keras.layers.Dense(8, activation='relu', kernel_regularizer=tf.keras.regularizers.l
    tf.keras.layers.Dense(1, activation='sigmoid', kernel_regularizer=tf.keras.regularizer
])

nn_with_reg.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

nn_with_reg.fit(X_train, y_train, epochs=200)
```

```
Epoch 1/200
6/6 [==============================] - 1s 2ms/step - loss: 1.4299 - accuracy: 0.4157
Epoch 2/200
6/6 [==============================] - 0s 2ms/step - loss: 1.2599 - accuracy: 0.4759
Epoch 3/200
6/6 [==============================] - 0s 2ms/step - loss: 1.1779 - accuracy: 0.5241
Epoch 4/200
6/6 [==============================] - 0s 2ms/step - loss: 1.1275 - accuracy: 0.5602
Epoch 5/200
6/6 [==============================] - 0s 3ms/step - loss: 1.0895 - accuracy: 0.5964
Epoch 6/200
6/6 [==============================] - 0s 2ms/step - loss: 1.0558 - accuracy: 0.6566
Epoch 7/200
6/6 [==============================] - 0s 2ms/step - loss: 1.0239 - accuracy: 0.6867
Epoch 8/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9964 - accuracy: 0.7169
Epoch 9/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9729 - accuracy: 0.7289
Epoch 10/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9520 - accuracy: 0.7530

<keras.callbacks.History at 0x7f8262a8c9d0>
```

```
%%time

nn_with_earlystopping = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32, input_dim=60, activation='relu'),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

early_stop = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=5, restore_best
nn_with_earlystopping.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accu

history = nn_with_earlystopping.fit(X_train, y_train, epochs=200, batch_size=8, validatior
```

```
Epoch 1/200
21/21 [==============================] - 1s 13ms/step - loss: 2.4736 - accuracy: 0.5120
Epoch 2/200
21/21 [==============================] - 0s 4ms/step - loss: 1.1687 - accuracy: 0.5301 
Epoch 3/200
21/21 [==============================] - 0s 3ms/step - loss: 0.7369 - accuracy: 0.6265 
Epoch 4/200
21/21 [==============================] - 0s 4ms/step - loss: 0.5668 - accuracy: 0.6687 
Epoch 5/200
21/21 [==============================] - 0s 4ms/step - loss: 0.4709 - accuracy: 0.7590 
Epoch 6/200
21/21 [==============================] - 0s 3ms/step - loss: 0.3951 - accuracy: 0.8253 
Epoch 7/200
21/21 [==============================] - 0s 3ms/step - loss: 0.3393 - accuracy: 0.8735 
Epoch 8/200
21/21 [==============================] - 0s 4ms/step - loss: 0.2908 - accuracy: 0.9217 
Epoch 9/200
21/21 [==============================] - 0s 3ms/step - loss: 0.2513 - accuracy: 0.9578 
Epoch 10/200
21/21 [==============================] - 0s 4ms/step - loss: 0.2176 - accuracy: 0.9759 
```

```
%%time
nn_with_dropout = tf.keras.models.Sequential([
    tf.keras.layers.Dense(32, input_dim=60, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(16, activation='relu'),
    tf.keras.layers.Dense(8, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(1, activation='sigmoid')
])


nn_with_dropout.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
nn_with_dropout.fit(X_train, y_train, epochs=200)
```

```
Epoch 1/200
6/6 [==============================] - 1s 2ms/step - loss: 1.9114 - accuracy: 0.5422
Epoch 2/200
6/6 [==============================] - 0s 2ms/step - loss: 1.8120 - accuracy: 0.5181
Epoch 3/200
6/6 [==============================] - 0s 2ms/step - loss: 1.0913 - accuracy: 0.5904
Epoch 4/200
6/6 [==============================] - 0s 3ms/step - loss: 1.1378 - accuracy: 0.5843
Epoch 5/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9297 - accuracy: 0.6145
Epoch 6/200
6/6 [==============================] - 0s 2ms/step - loss: 0.9452 - accuracy: 0.6265
Epoch 7/200
6/6 [==============================] - 0s 2ms/step - loss: 1.1592 - accuracy: 0.5783
Epoch 8/200
6/6 [==============================] - 0s 2ms/step - loss: 0.8549 - accuracy: 0.6446
Epoch 9/200
6/6 [==============================] - 0s 2ms/step - loss: 0.7766 - accuracy: 0.6145
Epoch 10/200
6/6 [==============================] - 0s 2ms/step - loss: 0.7642 - accuracy: 0.6867

<keras.callbacks.History at 0x7f825c7053d0>
```