DNN Lab

Objectives

- Understand basic DNN model building process using Keras
- Analyze model performance and capacity vs generalization tradeoff
- Modify models to reduce overfitting and improve performance

Exercises

- Build a DNN model for slump Test Problem
- Start with a model consisting of one hidden layer with 7 neurons
- Analyze results and explore improvements to model in terms of capacity, regularization

Step 1: Import Libraries

In [1]:
```
%tensorflow_version 2.x
from numpy.random import seed
seed(2)
import tensorflow as tf
from tensorflow import keras
from IPython import display
from matplotlib import cm
from matplotlib import gridspec
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd
import os
import datetime
from tensorflow.python.data import Dataset
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler, StandardScaler
from sklearn.model_selection import train_test_split
from collections import Counter
from imblearn.over_sampling import SMOTE, RandomOverSampler

print(tf.__version__)
```

```
2.6.0

/usr/local/lib/python3.7/dist-packages/sklearn/externals/six.py:31: FutureWar
ning: The module is deprecated in version 0.21 and will be removed in version
0.23 since we've dropped support for Python 2.7. Please rely on the official
version of six (https://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:144: Futu
reWarning: The sklearn.neighbors.base module is  deprecated in version 0.22 a
nd will be removed in version 0.24. The corresponding classes / functions sho
uld instead be imported from sklearn.neighbors. Anything that cannot be impor
ted from sklearn.neighbors is now part of the private API.
  warnings.warn(message, FutureWarning)
```

In [2]:
```
#we could remove the age and no longer need to standardize it, then rerun the
 model
```

Step 2: Import Data

In [3]:
```
pd.options.display.max_rows = 10
pd.options.display.float_format = '{:.4f}'.format

train_data = pd.read_csv("ticdata2000.txt", sep="\t",header=None)

train_data = train_data.reindex(
    np.random.permutation(train_data.index))
```

In [4]:
```
train_data.shape[0]
#train_data.head(10)
#train_data.iloc[:,43:65]
#train_data.iloc[:,84:87]
#train_data.iloc[[58]].columns.max()
#train_data[:][58].max()
train_data[:][55].min()
```

Out[4]: 0

Step 3: Preprocess

In [5]:
```
#checking which columns have NaN values
train_data[train_data.isnull().any(axis=1)]
#checking to see the # of NaN values present
len(train_data[train_data.isnull().any(axis=1)])
```

Out[5]: 0

In [6]:
```
OverSample= RandomOverSampler(random_state=42)
X = train_data.iloc[:,:-2]
y = train_data.iloc[:,85:]
# fit and apply the transform
X_oversample, y_oversample = OverSample.fit_resample(X,y)

print(f'Before Oversampling: {X.shape}')
print(f'After Oversampling: {X_oversample.shape}')
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:760: DataC
onversionWarning: A column-vector y was passed when a 1d array was expected.
Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

Before Oversampling: (5822, 84)
After Oversampling: (10948, 84)

/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: Futur
eWarning: Function safe_indexing is deprecated; safe_indexing is deprecated i
n version 0.22 and will be removed in version 0.24.
  warnings.warn(msg, category=FutureWarning)
```

In [7]:
```
y_df= pd.DataFrame(y_oversample)
X_df = pd.DataFrame(X_oversample)
```

In [8]:
```
#changing predictor variable into dummy vars
y_df.rename(columns = { 0 : 'Insurance_Purchased'}, inplace=True)
y_df= pd.get_dummies(y_df, columns=['Insurance_Purchased'])
```

In [9]:
```
#Baseline accuracy measure
naive_app_min= y_df['Insurance_Purchased_0'].value_counts().max()/len(y_df)
naive_app_min
```

Out[9]: 0.5

Train/Validation Split

In [10]:
```
#Creating a training and validation dataset with a 80/20 split
X_train,X_test, y_train, y_test = train_test_split(X_df,y_df, test_size=0.2, r
andom_state=1)
```

In [11]:
```
X_train.head()
```

Out[11]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
10414	6	1	3	2	2	0	4	2	4	9	0	0	0	5	4	3	5	2	7	0	0	2	0	0
4736	38	1	4	2	9	0	4	0	5	8	0	1	1	1	8	1	4	5	1	0	0	2	0	7
5240	33	2	3	3	8	0	4	0	5	7	0	2	2	1	6	0	2	7	0	0	0	3	6	0
1576	33	1	2	4	8	0	5	0	5	9	0	0	0	9	0	0	9	0	0	0	0	9	0	0
6337	4	2	2	5	1	0	7	0	2	7	0	2	2	7	0	2	6	2	6	0	0	3	0	0

5 rows × 84 columns

Step 4: Build Model

https://www.tensorflow.org/api_docs/python/tf/keras/Model (https://www.tensorflow.org/api_docs/python/tf/keras/Model)

https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense (https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense)

https://keras.io/optimizers/ (https://keras.io/optimizers/)

In [12]:
```
#Standardizing training dataset
#scaler = StandardScaler()

#scaledf = scaler.fit_transform(X_train)
#X_train = pd.DataFrame(scaledf, index=X_train.index, columns=X_train.columns)

print(X_train)
#Standardizing validation dataset
#vscaled = scaler.transform(X_test.values)
#X_test = pd.DataFrame(vscaled, index=X_test.index, columns=X_test.columns)
#print(X_test)
```

```
       0   1   2   3   4   5   6   7   8   ...  75  76  77  78  79  80  81  8
2  83
10414   6   1   3   2   2   0   4   2   4  ...   0   0   0   0   0   0   0
0   0
4736   38   1   4   2   9   0   4   0   5  ...   0   0   0   0   1   0   0
0   0
5240   33   2   3   3   8   0   4   0   5  ...   0   0   0   0   0   0   0
1   0
1576   33   1   2   4   8   0   5   0   5  ...   0   0   0   0   0   0   0
1   0
6337    4   2   2   5   1   0   7   0   2  ...   0   0   0   0   0   0   0
0   0
...    ..  ..  ..  ..  ..  ..  ..  ..  ..  ...  ..  ..  ..  ..  ..  ..  ..
..  ..
2895   33   2   4   2   8   0   5   1   4  ...   0   0   0   0   0   0   0
0   0
7813   12   2   4   2   3   2   4   2   2  ...   0   0   1   0   1   0   0
0   0
905    39   1   3   3   9   0   7   0   2  ...   0   0   0   0   1   0   0
0   0
5192   41   1   4   2  10   1   5   1   3  ...   0   0   0   0   1   0   0
0   0
235    33   1   2   3   8   0   7   0   2  ...   0   0   0   0   1   0   0
0   0

[8758 rows x 84 columns]
```

Build Model

In [13]:
```
l2_model = keras.Sequential([
    keras.layers.Dense(86, activation=tf.nn.relu,
                       input_shape=(X_train.shape[1],)),
    keras.layers.Dropout(0.75),
    keras.layers.Dense(40, activation=tf.nn.leaky_relu),
    keras.layers.Dropout(0.75),
    keras.layers.Dense(20, activation=tf.nn.sigmoid),
    keras.layers.Dropout(0.75),
    keras.layers.Dense(8, activation=tf.nn.leaky_relu),
    keras.layers.Dropout(0.75),
    keras.layers.Dense(3, activation=tf.nn.sigmoid),
    keras.layers.Dropout(0.75),
    keras.layers.Dense(2, activation=tf.nn.sigmoid)
  ])

#optimizer = tf.keras.optimizers.RMSprop(0.001)
optimizer = tf.keras.optimizers.Adam()

l2_model.compile(loss=tf.keras.losses.BinaryCrossentropy(),
                 optimizer='sgd',
                 metrics=['accuracy'])
l2_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 86)                7310
_________________________________________________________________
dropout (Dropout)            (None, 86)                0
_________________________________________________________________
dense_1 (Dense)              (None, 40)                3480
_________________________________________________________________
dropout_1 (Dropout)          (None, 40)                0
_________________________________________________________________
dense_2 (Dense)              (None, 20)                820
_________________________________________________________________
dropout_2 (Dropout)          (None, 20)                0
_________________________________________________________________
dense_3 (Dense)              (None, 8)                 168
_________________________________________________________________
dropout_3 (Dropout)          (None, 8)                 0
_________________________________________________________________
dense_4 (Dense)              (None, 3)                 27
_________________________________________________________________
dropout_4 (Dropout)          (None, 3)                 0
_________________________________________________________________
dense_5 (Dense)              (None, 2)                 8
=================================================================
Total params: 11,813
Trainable params: 11,813
Non-trainable params: 0
_________________________________________________________________
```

Fit Model

In [14]:
```
class PrintDot(keras.callbacks.Callback):
  def on_epoch_end(self, epoch, logs):
    if epoch % 100 == 0: print('')
    print('.', end='')

EPOCHS = 200

# Store training stats
l2_history = l2_model.fit(X_train, y_train, epochs=EPOCHS,
                    validation_data= (X_test, y_test),verbose=1)
```

```
Epoch 1/200
274/274 [==============================] - 2s 4ms/step - loss: 0.8813 - accur
acy: 0.4934 - val_loss: 0.6989 - val_accuracy: 0.5009
Epoch 2/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7923 - accur
acy: 0.4961 - val_loss: 0.6957 - val_accuracy: 0.4845
Epoch 3/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7555 - accur
acy: 0.4954 - val_loss: 0.6946 - val_accuracy: 0.4767
Epoch 4/200
274/274 [==============================] - 1s 3ms/step - loss: 0.7254 - accur
acy: 0.5037 - val_loss: 0.6941 - val_accuracy: 0.4918
Epoch 5/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7150 - accur
acy: 0.5042 - val_loss: 0.6938 - val_accuracy: 0.4904
Epoch 6/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7099 - accur
acy: 0.4904 - val_loss: 0.6937 - val_accuracy: 0.5014
Epoch 7/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7041 - accur
acy: 0.5039 - val_loss: 0.6936 - val_accuracy: 0.5009
Epoch 8/200
274/274 [==============================] - 1s 2ms/step - loss: 0.7003 - accur
acy: 0.5026 - val_loss: 0.6936 - val_accuracy: 0.5009
Epoch 9/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6977 - accur
acy: 0.5018 - val_loss: 0.6935 - val_accuracy: 0.5009
Epoch 10/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6967 - accur
acy: 0.4982 - val_loss: 0.6935 - val_accuracy: 0.5009
Epoch 11/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6949 - accur
acy: 0.5040 - val_loss: 0.6935 - val_accuracy: 0.5009
Epoch 12/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6950 - accur
acy: 0.5016 - val_loss: 0.6935 - val_accuracy: 0.5009
Epoch 13/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6950 - accur
acy: 0.4979 - val_loss: 0.6935 - val_accuracy: 0.5009
Epoch 14/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6953 - accur
acy: 0.4953 - val_loss: 0.6934 - val_accuracy: 0.5009
Epoch 15/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6941 - accur
acy: 0.4983 - val_loss: 0.6934 - val_accuracy: 0.5009
Epoch 16/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6939 - accur
acy: 0.4974 - val_loss: 0.6934 - val_accuracy: 0.5009
Epoch 17/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6944 - accur
acy: 0.4954 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 18/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6937 - accur
acy: 0.4979 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 19/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6934 - accur
acy: 0.5034 - val_loss: 0.6933 - val_accuracy: 0.5009
```

```
Epoch 20/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6937 - accur
acy: 0.4995 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 21/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6935 - accur
acy: 0.5013 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 22/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5089 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 23/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6934 - accur
acy: 0.5006 - val_loss: 0.6933 - val_accuracy: 0.5009
Epoch 24/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6941 - accur
acy: 0.4872 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 25/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.5013 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 26/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5033 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 27/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5018 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 28/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5031 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 29/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6933 - accur
acy: 0.5010 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 30/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5046 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 31/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5029 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 32/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4975 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 33/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5039 - val_loss: 0.6932 - val_accuracy: 0.4338
Epoch 34/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4997 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 35/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4989 - val_loss: 0.6932 - val_accuracy: 0.4840
Epoch 36/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4965 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 37/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.4993 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 38/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4950 - val_loss: 0.6931 - val_accuracy: 0.5009
```

```
Epoch 39/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4954 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 40/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6932 - val_accuracy: 0.4356
Epoch 41/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5038 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 42/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4973 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 43/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5088 - val_loss: 0.6932 - val_accuracy: 0.4945
Epoch 44/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5001 - val_loss: 0.6932 - val_accuracy: 0.5027
Epoch 45/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6930 - accur
acy: 0.5082 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 46/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4982 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 47/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.5030 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 48/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4970 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 49/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.4953 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 50/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5011 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 51/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4822 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 52/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5032 - val_loss: 0.6932 - val_accuracy: 0.5000
Epoch 53/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4954 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 54/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4973 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 55/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4991 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 56/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4881 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 57/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4936 - val_loss: 0.6932 - val_accuracy: 0.4991
```

```
Epoch 58/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4927 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 59/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4945 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 60/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4950 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 61/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4977 - val_loss: 0.6932 - val_accuracy: 0.4726
Epoch 62/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5025 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 63/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4957 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 64/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4944 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 65/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4938 - val_loss: 0.6931 - val_accuracy: 0.5196
Epoch 66/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4984 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 67/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4931 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 68/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5010 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 69/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4917 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 70/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.5022 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 71/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4976 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 72/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4979 - val_loss: 0.6932 - val_accuracy: 0.4973
Epoch 73/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5026 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 74/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5026 - val_loss: 0.6932 - val_accuracy: 0.4726
Epoch 75/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4978 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 76/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4954 - val_loss: 0.6931 - val_accuracy: 0.5009
```

```
Epoch 77/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4945 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 78/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5016 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 79/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4946 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 80/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4981 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 81/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4962 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 82/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4960 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 83/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 84/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4998 - val_loss: 0.6931 - val_accuracy: 0.5023
Epoch 85/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4923 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 86/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4913 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 87/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4993 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 88/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5000 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 89/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4971 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 90/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4997 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 91/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4963 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 92/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5031 - val_loss: 0.6932 - val_accuracy: 0.4941
Epoch 93/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.4962 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 94/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4999 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 95/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4985 - val_loss: 0.6932 - val_accuracy: 0.4849
```

```
Epoch 96/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6930 - accur
acy: 0.5064 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 97/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6934 - accur
acy: 0.4981 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 98/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6931 - accur
acy: 0.5022 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 99/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.5007 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 100/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6933 - accur
acy: 0.4965 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 101/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5042 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 102/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4936 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 103/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4960 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 104/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4955 - val_loss: 0.6932 - val_accuracy: 0.4507
Epoch 105/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4917 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 106/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4974 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 107/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4970 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 108/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4979 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 109/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5024 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 110/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4944 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 111/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4969 - val_loss: 0.6931 - val_accuracy: 0.5543
Epoch 112/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4949 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 113/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5038 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 114/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4933 - val_loss: 0.6931 - val_accuracy: 0.4991
```

```
Epoch 115/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5080 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 116/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.5018 - val_loss: 0.6931 - val_accuracy: 0.5037
Epoch 117/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4958 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 118/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5016 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 119/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5027 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 120/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4931 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 121/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5019 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 122/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4918 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 123/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5022 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 124/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4946 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 125/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4847 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 126/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5035 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 127/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4929 - val_loss: 0.6931 - val_accuracy: 0.5530
Epoch 128/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4929 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 129/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4949 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 130/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 131/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4898 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 132/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4965 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 133/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4963 - val_loss: 0.6931 - val_accuracy: 0.4922
```

```
Epoch 134/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4939 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 135/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4876 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 136/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4975 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 137/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4989 - val_loss: 0.6931 - val_accuracy: 0.5160
Epoch 138/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4928 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 139/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4982 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 140/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4969 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 141/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4953 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 142/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5032 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 143/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5002 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 144/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4997 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 145/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 146/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6933 - accur
acy: 0.4968 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 147/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.5058 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 148/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5095 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 149/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4912 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 150/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4989 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 151/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5069 - val_loss: 0.6931 - val_accuracy: 0.5041
Epoch 152/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6931 - val_accuracy: 0.4991
```

```
Epoch 153/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6931 - accur
acy: 0.5024 - val_loss: 0.6931 - val_accuracy: 0.5151
Epoch 154/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5030 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 155/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5002 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 156/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4984 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 157/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4977 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 158/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4953 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 159/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4973 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 160/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.4978 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 161/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4975 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 162/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4999 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 163/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4882 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 164/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4944 - val_loss: 0.6932 - val_accuracy: 0.4466
Epoch 165/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4922 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 166/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4906 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 167/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4947 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 168/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4898 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 169/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5053 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 170/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4968 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 171/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4946 - val_loss: 0.6931 - val_accuracy: 0.5260
```

```
Epoch 172/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4944 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 173/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.5017 - val_loss: 0.6931 - val_accuracy: 0.4986
Epoch 174/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4969 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 175/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4921 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 176/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4975 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 177/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4979 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 178/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5046 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 179/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5022 - val_loss: 0.6931 - val_accuracy: 0.4991
Epoch 180/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5042 - val_loss: 0.6931 - val_accuracy: 0.5068
Epoch 181/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6933 - accur
acy: 0.4951 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 182/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4974 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 183/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4952 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 184/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5034 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 185/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6932 - accur
acy: 0.4925 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 186/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4929 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 187/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5039 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 188/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5016 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 189/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4951 - val_loss: 0.6932 - val_accuracy: 0.5009
Epoch 190/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6931 - accur
acy: 0.4981 - val_loss: 0.6931 - val_accuracy: 0.5009
```

```
Epoch 191/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4930 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 192/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6931 - accur
acy: 0.5016 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 193/200
274/274 [==============================] - 1s 3ms/step - loss: 0.6931 - accur
acy: 0.5100 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 194/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.4966 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 195/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4950 - val_loss: 0.6932 - val_accuracy: 0.4991
Epoch 196/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.4975 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 197/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6932 - accur
acy: 0.5054 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 198/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5027 - val_loss: 0.6931 - val_accuracy: 0.5009
Epoch 199/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6933 - accur
acy: 0.5002 - val_loss: 0.6931 - val_accuracy: 0.5416
Epoch 200/200
274/274 [==============================] - 1s 2ms/step - loss: 0.6931 - accur
acy: 0.5054 - val_loss: 0.6931 - val_accuracy: 0.5009
```

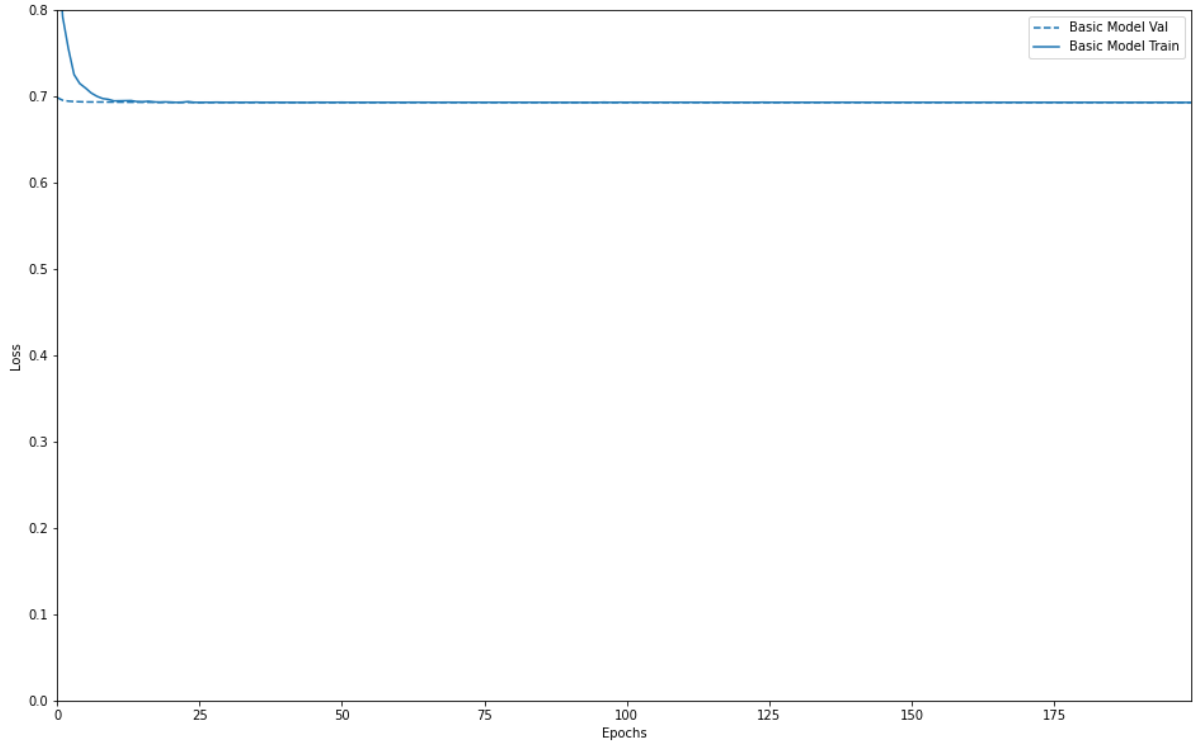
Lowest Validation Error

In [15]:
```
y_pred = np.round(l2_model.predict_on_batch(X_test),5)
print(y_pred)
```

```
[[0.49971 0.50029]
 [0.49965 0.50035]
 [0.49965 0.50035]
 ...
 [0.49968 0.50032]
 [0.49975 0.50025]
 [0.4997  0.5003 ]]
```

Step 5: Plot Results

In [16]:
```
import matplotlib.pyplot as plt

def plot_history(histories, key='loss'):
  plt.figure(figsize=(16,10))
  for name, history in histories:
    val = plt.plot(l2_history.epoch, l2_history.history['val_'+key],
                   '--', label=name.title()+' Val')
    plt.plot(l2_history.epoch, l2_history.history[key], color=val[0].get_color
(),
             label=name.title()+' Train')

  plt.xlabel('Epochs')
  plt.ylabel(key.replace('_',' ').title())
  plt.legend()

  plt.xlim([0,max(l2_history.epoch)])
  plt.ylim([0,0.8])

plot_history([('Basic Model', l2_history)])

#Plot Multiple Model Results

#plot_history([('Plain', m1_history),('L1',model1)])
```



In [17]:
```
print(max(l2_history.history['val_accuracy']))
```

```
0.5543379187583923
```

Goal: Predict whether the person purchased Caravan Insurance. Given the 2 categories we're trying to predict the a model >naive approach accuracy of 50% after implementing RandomOversampling. The data was very skewed 5474 to 348 prior to it.

As you can see in the code above the max val accuracy was 55.43%, which is slightly higher than the baseline accuracy of 50%.

Predictions