{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

},

"accelerator": "TPU"

},

"cells": [

{

"cell\_type": "markdown",

"source": [

"Assignment 1: Neural Networks"

],

"metadata": {

"id": "XQJ5edZBQGNe"

}

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "qJJjKIobPz-8"

},

"outputs": [],

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"from sklearn.model\_selection import train\_test\_split\n",

"import tensorflow as tf\n",

"from tensorflow import keras\n",

"from tensorflow.keras.models import Sequential\n",

"from tensorflow.keras.layers import Dense\n",

"from tensorflow.keras import layers\n",

"import matplotlib.pyplot as plt"

]

},

{

"cell\_type": "code",

"source": [

"imdb = keras.datasets.imdb"

],

"metadata": {

"id": "gw5hPX6nQQ8i"

},

"execution\_count": 2,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"(train\_data, train\_labels), (test\_data, test\_labels) = imdb.load\_data(num\_words=10000)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "DAKlHTerQmWd",

"outputId": "04d60d93-a906-4739-b409-2e2e05afe3c8"

},

"execution\_count": 3,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb.npz\n",

"17464789/17464789 [==============================] - 0s 0us/step\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"print(\"Training entries: {}, labels: {}\".format(len(train\_data), len(train\_labels)))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "dN-K\_ttYQuHi",

"outputId": "e3a31370-7d93-4b13-9b10-aae9d9a1e65e"

},

"execution\_count": 4,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Training entries: 25000, labels: 25000\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"word\_index=imdb.get\_word\_index()\n",

"reverse\_word\_index=dict(\n",

" [(value,key) for (key,value) in word\_index.items()])\n",

"decoded\_review = \"\".join(\n",

" [reverse\_word\_index.get(i-3,\"?\") for i in train\_data[0]])"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "FRMpID1QQxcI",

"outputId": "729094df-222b-4cbf-d959-f35bed42ba03"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/imdb\_word\_index.json\n",

"1641221/1641221 [==============================] - 0s 0us/step\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"def vectorize\_sequences(sequences, dimension=10000):\n",

" results=np.zeros((len(sequences),dimension))\n",

" for i,sequence in enumerate(sequences):\n",

" for j in sequence:\n",

" results[i,j]=1.\n",

" return results\n",

"x\_train=vectorize\_sequences(train\_data)\n",

"x\_test=vectorize\_sequences(test\_data)"

],

"metadata": {

"id": "I7S07fCqQ0hL"

},

"execution\_count": 6,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"x\_train[0]"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "4xzEIRLrQ3W1",

"outputId": "77ad5ec8-4700-43e1-de5f-7de86ec75797"

},

"execution\_count": 7,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([0., 1., 1., ..., 0., 0., 0.])"

]

},

"metadata": {},

"execution\_count": 7

}

]

},

{

"cell\_type": "code",

"source": [

"y\_train=np.asarray(train\_labels).astype(\"float32\")\n",

"y\_test=np.asarray(test\_labels).astype(\"float32\")"

],

"metadata": {

"id": "8MjU1xigQ6Ol"

},

"execution\_count": 8,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model=keras.Sequential([\n",

" layers.Dense(16, activation=\"relu\"),\n",

" layers.Dense(16, activation = \"relu\"),\n",

" layers.Dense(1, activation=\"sigmoid\")\n",

"])"

],

"metadata": {

"id": "ai1FtiDGQ9bY"

},

"execution\_count": 9,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.compile(optimizer=\"adam\",\n",

" loss=\"binary\_crossentropy\",\n",

" metrics=[\"accuracy\"])\n"

],

"metadata": {

"id": "FA-fvvpcRAIk"

},

"execution\_count": 10,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"#creating validation set\n",

"x\_val=x\_train[:10000]\n",

"partial\_x\_train=x\_train[10000:]\n",

"y\_val=y\_train[:10000]\n",

"partial\_y\_train=y\_train[10000:]\n"

],

"metadata": {

"id": "g5JdUAcbREcl"

},

"execution\_count": 11,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"history= model.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "W8JfxmvpRImx",

"outputId": "40179cf2-fce8-4cd2-8477-827c2913b344"

},

"execution\_count": 12,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 4s 98ms/step - loss: 0.5546 - accuracy: 0.7721 - val\_loss: 0.4042 - val\_accuracy: 0.8549\n",

"Epoch 2/20\n",

"30/30 [==============================] - 2s 82ms/step - loss: 0.2993 - accuracy: 0.8993 - val\_loss: 0.2994 - val\_accuracy: 0.8825\n",

"Epoch 3/20\n",

"30/30 [==============================] - 2s 66ms/step - loss: 0.2022 - accuracy: 0.9337 - val\_loss: 0.2754 - val\_accuracy: 0.8903\n",

"Epoch 4/20\n",

"30/30 [==============================] - 2s 56ms/step - loss: 0.1482 - accuracy: 0.9536 - val\_loss: 0.2804 - val\_accuracy: 0.8865\n",

"Epoch 5/20\n",

"30/30 [==============================] - 2s 57ms/step - loss: 0.1125 - accuracy: 0.9685 - val\_loss: 0.3024 - val\_accuracy: 0.8825\n",

"Epoch 6/20\n",

"30/30 [==============================] - 2s 56ms/step - loss: 0.0862 - accuracy: 0.9790 - val\_loss: 0.3186 - val\_accuracy: 0.8831\n",

"Epoch 7/20\n",

"30/30 [==============================] - 1s 48ms/step - loss: 0.0651 - accuracy: 0.9862 - val\_loss: 0.3423 - val\_accuracy: 0.8781\n",

"Epoch 8/20\n",

"30/30 [==============================] - 1s 42ms/step - loss: 0.0489 - accuracy: 0.9912 - val\_loss: 0.3743 - val\_accuracy: 0.8796\n",

"Epoch 9/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0364 - accuracy: 0.9957 - val\_loss: 0.4023 - val\_accuracy: 0.8771\n",

"Epoch 10/20\n",

"30/30 [==============================] - 2s 74ms/step - loss: 0.0275 - accuracy: 0.9974 - val\_loss: 0.4268 - val\_accuracy: 0.8762\n",

"Epoch 11/20\n",

"30/30 [==============================] - 1s 48ms/step - loss: 0.0203 - accuracy: 0.9988 - val\_loss: 0.4596 - val\_accuracy: 0.8739\n",

"Epoch 12/20\n",

"30/30 [==============================] - 1s 45ms/step - loss: 0.0150 - accuracy: 0.9995 - val\_loss: 0.4912 - val\_accuracy: 0.8733\n",

"Epoch 13/20\n",

"30/30 [==============================] - 2s 56ms/step - loss: 0.0111 - accuracy: 0.9998 - val\_loss: 0.5185 - val\_accuracy: 0.8713\n",

"Epoch 14/20\n",

"30/30 [==============================] - 1s 47ms/step - loss: 0.0085 - accuracy: 0.9999 - val\_loss: 0.5467 - val\_accuracy: 0.8706\n",

"Epoch 15/20\n",

"30/30 [==============================] - 2s 53ms/step - loss: 0.0066 - accuracy: 0.9999 - val\_loss: 0.5724 - val\_accuracy: 0.8697\n",

"Epoch 16/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 0.0053 - accuracy: 0.9999 - val\_loss: 0.5950 - val\_accuracy: 0.8683\n",

"Epoch 17/20\n",

"30/30 [==============================] - 2s 66ms/step - loss: 0.0043 - accuracy: 0.9999 - val\_loss: 0.6163 - val\_accuracy: 0.8679\n",

"Epoch 18/20\n",

"30/30 [==============================] - 2s 76ms/step - loss: 0.0035 - accuracy: 0.9999 - val\_loss: 0.6370 - val\_accuracy: 0.8685\n",

"Epoch 19/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 0.0030 - accuracy: 1.0000 - val\_loss: 0.6553 - val\_accuracy: 0.8673\n",

"Epoch 20/20\n",

"30/30 [==============================] - 2s 50ms/step - loss: 0.0025 - accuracy: 1.0000 - val\_loss: 0.6731 - val\_accuracy: 0.8674\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"history\_dict=history.history\n",

"history\_dict.keys()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "DVGv8ccxRS1c",

"outputId": "05ee4e51-5c06-4019-ace4-641308e0d427"

},

"execution\_count": 13,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"dict\_keys(['loss', 'accuracy', 'val\_loss', 'val\_accuracy'])"

]

},

"metadata": {},

"execution\_count": 13

}

]

},

{

"cell\_type": "code",

"source": [

"# Extract loss values from the training history\n",

"train\_loss = history.history['loss']\n",

"val\_loss = history.history['val\_loss']\n",

"\n",

"# Create an array of epoch numbers\n",

"epochs = range(1, len(train\_loss) + 1)\n",

"\n",

"# Plot the training and validation loss\n",

"plt.plot(epochs, train\_loss, 'b', label='Training Loss')\n",

"plt.plot(epochs, val\_loss, 'r', label='Validation Loss')\n",

"plt.title('Training and Validation Loss')\n",

"plt.xlabel('Epochs')\n",

"plt.ylabel('Loss')\n",

"plt.legend()\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 472

},

"id": "n4rMr3RISKra",

"outputId": "6151b659-3cec-4207-e396-c926751f281e"

},

"execution\_count": 14,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"# Extract loss values from the training history\n",

"train\_accuracy = history.history['accuracy']\n",

"val\_accuracy = history.history['val\_accuracy']\n",

"\n",

"# Create an array of epoch numbers\n",

"epochs = range(1, len(train\_accuracy) + 1)\n",

"\n",

"# Plot the training and validation loss\n",

"plt.plot(epochs, train\_accuracy, 'b', label='Training Accuracy')\n",

"plt.plot(epochs, val\_accuracy, 'r', label='Validation Accuracy')\n",

"plt.title('Training and Validation Accuracy')\n",

"plt.xlabel('Epochs')\n",

"plt.ylabel('Accuracy')\n",

"plt.legend()\n",

"plt.show()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 472

},

"id": "ewK6OEO6SNMi",

"outputId": "1a5ece8b-9493-42c8-db9f-ecf5fcf8ca8b"

},

"execution\_count": 15,

"outputs": [

{

"output\_type": "display\_data",

"data": {

"text/plain": [

"<Figure size 640x480 with 1 Axes>"

],

"image/png": "\n"

},

"metadata": {}

}

]

},

{

"cell\_type": "code",

"source": [

"results=model.evaluate(x\_test,y\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "SqJF\_OYfSPnG",

"outputId": "72284205-7490-47ac-f476-72eabb0bc1ac"

},

"execution\_count": 16,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 2s 3ms/step - loss: 0.7275 - accuracy: 0.8558\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"model.predict(x\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "YsPn0Cq3SSNn",

"outputId": "f841226c-18a0-493d-d475-16777df68717"

},

"execution\_count": 17,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 2s 3ms/step\n"

]

},

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"array([[0.00594444],\n",

" [1. ],\n",

" [0.96707135],\n",

" ...,\n",

" [0.00110336],\n",

" [0.00661215],\n",

" [0.8637377 ]], dtype=float32)"

]

},

"metadata": {},

"execution\_count": 17

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 1.You used two hidden layers. Try using one or three hidden layers and see how doing so affects validation and test accuracy."

],

"metadata": {

"id": "KDf5xt15SbYk"

}

},

{

"cell\_type": "code",

"source": [

"model2=keras.Sequential([\n",

" layers.Dense(16, activation=\"relu\"),\n",

" layers.Dense(16, activation=\"relu\"),\n",

" layers.Dense(16, activation = \"relu\"),\n",

" layers.Dense(1, activation=\"sigmoid\")\n",

"])"

],

"metadata": {

"id": "nrtKy-26SYet"

},

"execution\_count": 18,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model2.compile(optimizer=\"adam\",\n",

" loss=\"binary\_crossentropy\",\n",

" metrics=[\"accuracy\"])\n",

"history2= model2.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "6H2I\_SskSheu",

"outputId": "0b4c7a06-23f2-4209-e977-040c22431eb9"

},

"execution\_count": 19,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 4s 97ms/step - loss: 0.6039 - accuracy: 0.7443 - val\_loss: 0.4590 - val\_accuracy: 0.8542\n",

"Epoch 2/20\n",

"30/30 [==============================] - 1s 47ms/step - loss: 0.3323 - accuracy: 0.8976 - val\_loss: 0.3049 - val\_accuracy: 0.8799\n",

"Epoch 3/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 0.1995 - accuracy: 0.9327 - val\_loss: 0.2787 - val\_accuracy: 0.8884\n",

"Epoch 4/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 0.1374 - accuracy: 0.9561 - val\_loss: 0.2956 - val\_accuracy: 0.8856\n",

"Epoch 5/20\n",

"30/30 [==============================] - 2s 57ms/step - loss: 0.0960 - accuracy: 0.9720 - val\_loss: 0.3198 - val\_accuracy: 0.8826\n",

"Epoch 6/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 0.0656 - accuracy: 0.9847 - val\_loss: 0.3583 - val\_accuracy: 0.8768\n",

"Epoch 7/20\n",

"30/30 [==============================] - 2s 72ms/step - loss: 0.0422 - accuracy: 0.9929 - val\_loss: 0.3951 - val\_accuracy: 0.8782\n",

"Epoch 8/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 0.0256 - accuracy: 0.9974 - val\_loss: 0.4370 - val\_accuracy: 0.8746\n",

"Epoch 9/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0157 - accuracy: 0.9989 - val\_loss: 0.4767 - val\_accuracy: 0.8731\n",

"Epoch 10/20\n",

"30/30 [==============================] - 1s 47ms/step - loss: 0.0100 - accuracy: 0.9997 - val\_loss: 0.5087 - val\_accuracy: 0.8714\n",

"Epoch 11/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0066 - accuracy: 0.9997 - val\_loss: 0.5398 - val\_accuracy: 0.8728\n",

"Epoch 12/20\n",

"30/30 [==============================] - 1s 49ms/step - loss: 0.0043 - accuracy: 0.9999 - val\_loss: 0.5670 - val\_accuracy: 0.8703\n",

"Epoch 13/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0031 - accuracy: 0.9999 - val\_loss: 0.5919 - val\_accuracy: 0.8706\n",

"Epoch 14/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 0.0024 - accuracy: 1.0000 - val\_loss: 0.6116 - val\_accuracy: 0.8702\n",

"Epoch 15/20\n",

"30/30 [==============================] - 3s 98ms/step - loss: 0.0019 - accuracy: 1.0000 - val\_loss: 0.6312 - val\_accuracy: 0.8697\n",

"Epoch 16/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0015 - accuracy: 1.0000 - val\_loss: 0.6475 - val\_accuracy: 0.8694\n",

"Epoch 17/20\n",

"30/30 [==============================] - 2s 59ms/step - loss: 0.0013 - accuracy: 1.0000 - val\_loss: 0.6628 - val\_accuracy: 0.8687\n",

"Epoch 18/20\n",

"30/30 [==============================] - 2s 55ms/step - loss: 0.0011 - accuracy: 1.0000 - val\_loss: 0.6780 - val\_accuracy: 0.8690\n",

"Epoch 19/20\n",

"30/30 [==============================] - 1s 46ms/step - loss: 9.3309e-04 - accuracy: 1.0000 - val\_loss: 0.6917 - val\_accuracy: 0.8688\n",

"Epoch 20/20\n",

"30/30 [==============================] - 2s 54ms/step - loss: 8.1115e-04 - accuracy: 1.0000 - val\_loss: 0.7044 - val\_accuracy: 0.8686\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"results2=model2.evaluate(x\_test,y\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "3Du24IEBSjqV",

"outputId": "e48d686e-d9a9-4a2e-e243-6414fc6176c3"

},

"execution\_count": 20,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 3s 4ms/step - loss: 0.7530 - accuracy: 0.8590\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 2. Try using layers with more hidden units or fewer hidden units: 32 units, 64 units, and so on."

],

"metadata": {

"id": "BIC9HfndCn4r"

}

},

{

"cell\_type": "code",

"source": [

"model3=keras.Sequential([\n",

" layers.Dense(32, activation=\"relu\"),\n",

" layers.Dense(32, activation = \"relu\"),\n",

" layers.Dense(1, activation=\"sigmoid\")\n",

"])"

],

"metadata": {

"id": "G\_BPXwiLSrwc"

},

"execution\_count": 21,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model3.compile(optimizer=\"adam\",\n",

" loss=\"binary\_crossentropy\",\n",

" metrics=[\"accuracy\"])\n",

"history3= model3.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "i2AuPvSXUVjm",

"outputId": "eb019a1f-a511-482b-af95-a678fe4d4623"

},

"execution\_count": 22,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 5s 102ms/step - loss: 0.5157 - accuracy: 0.7907 - val\_loss: 0.3410 - val\_accuracy: 0.8715\n",

"Epoch 2/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 0.2403 - accuracy: 0.9149 - val\_loss: 0.2768 - val\_accuracy: 0.8902\n",

"Epoch 3/20\n",

"30/30 [==============================] - 2s 70ms/step - loss: 0.1547 - accuracy: 0.9473 - val\_loss: 0.2874 - val\_accuracy: 0.8859\n",

"Epoch 4/20\n",

"30/30 [==============================] - 2s 64ms/step - loss: 0.1074 - accuracy: 0.9671 - val\_loss: 0.3193 - val\_accuracy: 0.8818\n",

"Epoch 5/20\n",

"30/30 [==============================] - 2s 67ms/step - loss: 0.0736 - accuracy: 0.9825 - val\_loss: 0.3494 - val\_accuracy: 0.8783\n",

"Epoch 6/20\n",

"30/30 [==============================] - 2s 76ms/step - loss: 0.0509 - accuracy: 0.9899 - val\_loss: 0.3873 - val\_accuracy: 0.8782\n",

"Epoch 7/20\n",

"30/30 [==============================] - 3s 86ms/step - loss: 0.0334 - accuracy: 0.9953 - val\_loss: 0.4332 - val\_accuracy: 0.8745\n",

"Epoch 8/20\n",

"30/30 [==============================] - 2s 68ms/step - loss: 0.0229 - accuracy: 0.9977 - val\_loss: 0.4735 - val\_accuracy: 0.8753\n",

"Epoch 9/20\n",

"30/30 [==============================] - 2s 64ms/step - loss: 0.0143 - accuracy: 0.9991 - val\_loss: 0.5101 - val\_accuracy: 0.8732\n",

"Epoch 10/20\n",

"30/30 [==============================] - 2s 67ms/step - loss: 0.0096 - accuracy: 0.9997 - val\_loss: 0.5456 - val\_accuracy: 0.8713\n",

"Epoch 11/20\n",

"30/30 [==============================] - 2s 67ms/step - loss: 0.0064 - accuracy: 0.9998 - val\_loss: 0.5765 - val\_accuracy: 0.8690\n",

"Epoch 12/20\n",

"30/30 [==============================] - 2s 76ms/step - loss: 0.0044 - accuracy: 1.0000 - val\_loss: 0.6023 - val\_accuracy: 0.8696\n",

"Epoch 13/20\n",

"30/30 [==============================] - 3s 86ms/step - loss: 0.0033 - accuracy: 1.0000 - val\_loss: 0.6243 - val\_accuracy: 0.8694\n",

"Epoch 14/20\n",

"30/30 [==============================] - 2s 69ms/step - loss: 0.0026 - accuracy: 1.0000 - val\_loss: 0.6450 - val\_accuracy: 0.8689\n",

"Epoch 15/20\n",

"30/30 [==============================] - 2s 63ms/step - loss: 0.0021 - accuracy: 1.0000 - val\_loss: 0.6632 - val\_accuracy: 0.8690\n",

"Epoch 16/20\n",

"30/30 [==============================] - 2s 69ms/step - loss: 0.0017 - accuracy: 1.0000 - val\_loss: 0.6796 - val\_accuracy: 0.8681\n",

"Epoch 17/20\n",

"30/30 [==============================] - 2s 70ms/step - loss: 0.0014 - accuracy: 1.0000 - val\_loss: 0.6946 - val\_accuracy: 0.8673\n",

"Epoch 18/20\n",

"30/30 [==============================] - 3s 87ms/step - loss: 0.0012 - accuracy: 1.0000 - val\_loss: 0.7083 - val\_accuracy: 0.8676\n",

"Epoch 19/20\n",

"30/30 [==============================] - 7s 224ms/step - loss: 0.0011 - accuracy: 1.0000 - val\_loss: 0.7214 - val\_accuracy: 0.8675\n",

"Epoch 20/20\n",

"30/30 [==============================] - 3s 110ms/step - loss: 9.2179e-04 - accuracy: 1.0000 - val\_loss: 0.7344 - val\_accuracy: 0.8671\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"results3=model3.evaluate(x\_test,y\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "fLX4BcRkUX5P",

"outputId": "6b55af69-b53e-4b54-a59b-9db270aec342"

},

"execution\_count": 23,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 3s 4ms/step - loss: 0.7907 - accuracy: 0.8567\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 3. Try using the mse loss function instead of binary\_crossentropy."

],

"metadata": {

"id": "z3bDlZCzC5If"

}

},

{

"cell\_type": "code",

"source": [

"model.compile(optimizer=\"adam\",\n",

" loss=\"mse\",\n",

" metrics=[\"accuracy\"])\n",

"history4= model.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "waeRlJtnUdqV",

"outputId": "41b8c257-1c18-40fc-9ff7-24aca042da18"

},

"execution\_count": 24,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 4s 87ms/step - loss: 2.0590e-04 - accuracy: 0.9999 - val\_loss: 0.1160 - val\_accuracy: 0.8659\n",

"Epoch 2/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 1.9826e-04 - accuracy: 0.9999 - val\_loss: 0.1171 - val\_accuracy: 0.8646\n",

"Epoch 3/20\n",

"30/30 [==============================] - 2s 58ms/step - loss: 1.5706e-04 - accuracy: 1.0000 - val\_loss: 0.1179 - val\_accuracy: 0.8638\n",

"Epoch 4/20\n",

"30/30 [==============================] - 2s 58ms/step - loss: 8.0067e-05 - accuracy: 1.0000 - val\_loss: 0.1191 - val\_accuracy: 0.8641\n",

"Epoch 5/20\n",

"30/30 [==============================] - 3s 84ms/step - loss: 4.3442e-05 - accuracy: 1.0000 - val\_loss: 0.1197 - val\_accuracy: 0.8640\n",

"Epoch 6/20\n",

"30/30 [==============================] - 1s 50ms/step - loss: 2.2530e-05 - accuracy: 1.0000 - val\_loss: 0.1202 - val\_accuracy: 0.8632\n",

"Epoch 7/20\n",

"30/30 [==============================] - 1s 49ms/step - loss: 1.4919e-05 - accuracy: 1.0000 - val\_loss: 0.1204 - val\_accuracy: 0.8621\n",

"Epoch 8/20\n",

"30/30 [==============================] - 2s 58ms/step - loss: 1.2111e-05 - accuracy: 1.0000 - val\_loss: 0.1205 - val\_accuracy: 0.8618\n",

"Epoch 9/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 1.0152e-05 - accuracy: 1.0000 - val\_loss: 0.1207 - val\_accuracy: 0.8623\n",

"Epoch 10/20\n",

"30/30 [==============================] - 2s 59ms/step - loss: 8.8359e-06 - accuracy: 1.0000 - val\_loss: 0.1207 - val\_accuracy: 0.8621\n",

"Epoch 11/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 7.8295e-06 - accuracy: 1.0000 - val\_loss: 0.1208 - val\_accuracy: 0.8622\n",

"Epoch 12/20\n",

"30/30 [==============================] - 2s 70ms/step - loss: 7.0037e-06 - accuracy: 1.0000 - val\_loss: 0.1209 - val\_accuracy: 0.8618\n",

"Epoch 13/20\n",

"30/30 [==============================] - 2s 67ms/step - loss: 6.3278e-06 - accuracy: 1.0000 - val\_loss: 0.1210 - val\_accuracy: 0.8621\n",

"Epoch 14/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 5.7782e-06 - accuracy: 1.0000 - val\_loss: 0.1210 - val\_accuracy: 0.8619\n",

"Epoch 15/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 5.2705e-06 - accuracy: 1.0000 - val\_loss: 0.1211 - val\_accuracy: 0.8621\n",

"Epoch 16/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 4.8376e-06 - accuracy: 1.0000 - val\_loss: 0.1212 - val\_accuracy: 0.8623\n",

"Epoch 17/20\n",

"30/30 [==============================] - 2s 59ms/step - loss: 4.4694e-06 - accuracy: 1.0000 - val\_loss: 0.1213 - val\_accuracy: 0.8623\n",

"Epoch 18/20\n",

"30/30 [==============================] - 1s 43ms/step - loss: 4.1370e-06 - accuracy: 1.0000 - val\_loss: 0.1213 - val\_accuracy: 0.8626\n",

"Epoch 19/20\n",

"30/30 [==============================] - 2s 64ms/step - loss: 3.8381e-06 - accuracy: 1.0000 - val\_loss: 0.1214 - val\_accuracy: 0.8623\n",

"Epoch 20/20\n",

"30/30 [==============================] - 2s 74ms/step - loss: 3.5826e-06 - accuracy: 1.0000 - val\_loss: 0.1214 - val\_accuracy: 0.8621\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 4. Try using the tanh activation (an activation that was popular in the early days of neural networks) instead of relu."

],

"metadata": {

"id": "YFcyWPHPD1U7"

}

},

{

"cell\_type": "code",

"source": [

"model4=keras.Sequential([\n",

" layers.Dense(16, activation=\"tanh\"),\n",

" layers.Dense(16, activation = \"tanh\"),\n",

" layers.Dense(1, activation=\"sigmoid\")\n",

"])"

],

"metadata": {

"id": "RrYZiLX-Ui7j"

},

"execution\_count": 25,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model4.compile(optimizer=\"adam\",\n",

" loss=\"binary\_crossentropy\",\n",

" metrics=[\"accuracy\"])\n",

"history4= model4.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "2cOeYqjbUmv\_",

"outputId": "47d65a0a-acc9-439f-faab-2e80c6253e1b"

},

"execution\_count": 26,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 4s 87ms/step - loss: 0.5116 - accuracy: 0.7893 - val\_loss: 0.3711 - val\_accuracy: 0.8598\n",

"Epoch 2/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 0.2716 - accuracy: 0.9076 - val\_loss: 0.2840 - val\_accuracy: 0.8875\n",

"Epoch 3/20\n",

"30/30 [==============================] - 2s 72ms/step - loss: 0.1816 - accuracy: 0.9407 - val\_loss: 0.2720 - val\_accuracy: 0.8886\n",

"Epoch 4/20\n",

"30/30 [==============================] - 1s 45ms/step - loss: 0.1309 - accuracy: 0.9623 - val\_loss: 0.2846 - val\_accuracy: 0.8860\n",

"Epoch 5/20\n",

"30/30 [==============================] - 1s 50ms/step - loss: 0.0964 - accuracy: 0.9756 - val\_loss: 0.3031 - val\_accuracy: 0.8830\n",

"Epoch 6/20\n",

"30/30 [==============================] - 1s 42ms/step - loss: 0.0717 - accuracy: 0.9843 - val\_loss: 0.3297 - val\_accuracy: 0.8786\n",

"Epoch 7/20\n",

"30/30 [==============================] - 1s 48ms/step - loss: 0.0521 - accuracy: 0.9907 - val\_loss: 0.3579 - val\_accuracy: 0.8765\n",

"Epoch 8/20\n",

"30/30 [==============================] - 2s 55ms/step - loss: 0.0377 - accuracy: 0.9946 - val\_loss: 0.3884 - val\_accuracy: 0.8736\n",

"Epoch 9/20\n",

"30/30 [==============================] - 1s 44ms/step - loss: 0.0271 - accuracy: 0.9974 - val\_loss: 0.4168 - val\_accuracy: 0.8727\n",

"Epoch 10/20\n",

"30/30 [==============================] - 2s 54ms/step - loss: 0.0198 - accuracy: 0.9987 - val\_loss: 0.4431 - val\_accuracy: 0.8700\n",

"Epoch 11/20\n",

"30/30 [==============================] - 2s 72ms/step - loss: 0.0147 - accuracy: 0.9995 - val\_loss: 0.4664 - val\_accuracy: 0.8698\n",

"Epoch 12/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 0.0110 - accuracy: 0.9997 - val\_loss: 0.4894 - val\_accuracy: 0.8682\n",

"Epoch 13/20\n",

"30/30 [==============================] - 1s 48ms/step - loss: 0.0085 - accuracy: 0.9999 - val\_loss: 0.5091 - val\_accuracy: 0.8684\n",

"Epoch 14/20\n",

"30/30 [==============================] - 1s 46ms/step - loss: 0.0068 - accuracy: 0.9999 - val\_loss: 0.5276 - val\_accuracy: 0.8674\n",

"Epoch 15/20\n",

"30/30 [==============================] - 1s 49ms/step - loss: 0.0057 - accuracy: 0.9999 - val\_loss: 0.5431 - val\_accuracy: 0.8678\n",

"Epoch 16/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 0.0048 - accuracy: 1.0000 - val\_loss: 0.5598 - val\_accuracy: 0.8656\n",

"Epoch 17/20\n",

"30/30 [==============================] - 2s 56ms/step - loss: 0.0040 - accuracy: 1.0000 - val\_loss: 0.5727 - val\_accuracy: 0.8670\n",

"Epoch 18/20\n",

"30/30 [==============================] - 2s 64ms/step - loss: 0.0035 - accuracy: 1.0000 - val\_loss: 0.5850 - val\_accuracy: 0.8675\n",

"Epoch 19/20\n",

"30/30 [==============================] - 2s 63ms/step - loss: 0.0030 - accuracy: 1.0000 - val\_loss: 0.5975 - val\_accuracy: 0.8659\n",

"Epoch 20/20\n",

"30/30 [==============================] - 2s 55ms/step - loss: 0.0027 - accuracy: 1.0000 - val\_loss: 0.6079 - val\_accuracy: 0.8664\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"results4=model4.evaluate(x\_test,y\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "9oSswqDwUpGT",

"outputId": "35623e25-a130-44b3-e617-ce7c1e37eca3"

},

"execution\_count": 27,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 2s 3ms/step - loss: 0.6603 - accuracy: 0.8550\n"

]

}

]

},

{

"cell\_type": "markdown",

"source": [

"## 5. Use any technique we studied in class, and these include regularization, dropout, etc., to get your model to perform better on validation."

],

"metadata": {

"id": "qOB0-lWlEM1N"

}

},

{

"cell\_type": "code",

"source": [

"model5=keras.Sequential([\n",

" layers.Dense(16, activation=\"relu\"),\n",

" layers.Dropout(0.2),\n",

" layers.Dense(16, activation = \"relu\"),\n",

" layers.Dropout(0.2),\n",

" layers.Dense(1, activation=\"sigmoid\")\n",

"])"

],

"metadata": {

"id": "3BG4ZUCGUyJH"

},

"execution\_count": 28,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model5.compile(optimizer=\"adam\",\n",

" loss=\"binary\_crossentropy\",\n",

" metrics=[\"accuracy\"])\n",

"history5= model5.fit(partial\_x\_train,\n",

" partial\_y\_train,\n",

" epochs=20,\n",

" batch\_size=512,\n",

" validation\_data=(x\_val,y\_val))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Q3Sz5W96UzO7",

"outputId": "55cf42b1-ef07-4511-845a-4a2d778c50f8"

},

"execution\_count": 29,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/20\n",

"30/30 [==============================] - 3s 75ms/step - loss: 0.6213 - accuracy: 0.6649 - val\_loss: 0.4995 - val\_accuracy: 0.8402\n",

"Epoch 2/20\n",

"30/30 [==============================] - 2s 55ms/step - loss: 0.4463 - accuracy: 0.8314 - val\_loss: 0.3601 - val\_accuracy: 0.8772\n",

"Epoch 3/20\n",

"30/30 [==============================] - 1s 43ms/step - loss: 0.3239 - accuracy: 0.8873 - val\_loss: 0.2980 - val\_accuracy: 0.8880\n",

"Epoch 4/20\n",

"30/30 [==============================] - 2s 64ms/step - loss: 0.2484 - accuracy: 0.9158 - val\_loss: 0.2788 - val\_accuracy: 0.8915\n",

"Epoch 5/20\n",

"30/30 [==============================] - 2s 69ms/step - loss: 0.1981 - accuracy: 0.9361 - val\_loss: 0.2783 - val\_accuracy: 0.8879\n",

"Epoch 6/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 0.1596 - accuracy: 0.9501 - val\_loss: 0.2904 - val\_accuracy: 0.8879\n",

"Epoch 7/20\n",

"30/30 [==============================] - 2s 61ms/step - loss: 0.1284 - accuracy: 0.9605 - val\_loss: 0.3039 - val\_accuracy: 0.8868\n",

"Epoch 8/20\n",

"30/30 [==============================] - 2s 59ms/step - loss: 0.1023 - accuracy: 0.9716 - val\_loss: 0.3233 - val\_accuracy: 0.8850\n",

"Epoch 9/20\n",

"30/30 [==============================] - 2s 52ms/step - loss: 0.0844 - accuracy: 0.9780 - val\_loss: 0.3532 - val\_accuracy: 0.8797\n",

"Epoch 10/20\n",

"30/30 [==============================] - 2s 62ms/step - loss: 0.0655 - accuracy: 0.9832 - val\_loss: 0.3789 - val\_accuracy: 0.8826\n",

"Epoch 11/20\n",

"30/30 [==============================] - 2s 77ms/step - loss: 0.0567 - accuracy: 0.9867 - val\_loss: 0.3952 - val\_accuracy: 0.8792\n",

"Epoch 12/20\n",

"30/30 [==============================] - 2s 60ms/step - loss: 0.0466 - accuracy: 0.9899 - val\_loss: 0.4184 - val\_accuracy: 0.8793\n",

"Epoch 13/20\n",

"30/30 [==============================] - 2s 59ms/step - loss: 0.0390 - accuracy: 0.9910 - val\_loss: 0.4471 - val\_accuracy: 0.8791\n",

"Epoch 14/20\n",

"30/30 [==============================] - 1s 49ms/step - loss: 0.0311 - accuracy: 0.9935 - val\_loss: 0.4841 - val\_accuracy: 0.8759\n",

"Epoch 15/20\n",

"30/30 [==============================] - 2s 57ms/step - loss: 0.0251 - accuracy: 0.9952 - val\_loss: 0.4990 - val\_accuracy: 0.8771\n",

"Epoch 16/20\n",

"30/30 [==============================] - 1s 47ms/step - loss: 0.0244 - accuracy: 0.9947 - val\_loss: 0.5040 - val\_accuracy: 0.8766\n",

"Epoch 17/20\n",

"30/30 [==============================] - 1s 45ms/step - loss: 0.0200 - accuracy: 0.9959 - val\_loss: 0.5491 - val\_accuracy: 0.8741\n",

"Epoch 18/20\n",

"30/30 [==============================] - 2s 54ms/step - loss: 0.0173 - accuracy: 0.9969 - val\_loss: 0.5661 - val\_accuracy: 0.8750\n",

"Epoch 19/20\n",

"30/30 [==============================] - 2s 76ms/step - loss: 0.0154 - accuracy: 0.9970 - val\_loss: 0.5914 - val\_accuracy: 0.8737\n",

"Epoch 20/20\n",

"30/30 [==============================] - 2s 51ms/step - loss: 0.0138 - accuracy: 0.9977 - val\_loss: 0.6210 - val\_accuracy: 0.8740\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"results=model5.evaluate(x\_test,y\_test)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "o-n2tf2zU1EC",

"outputId": "39a2d834-6645-454a-a177-384cfb421383"

},

"execution\_count": 30,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"782/782 [==============================] - 3s 4ms/step - loss: 0.6797 - accuracy: 0.8620\n"

]

}

]

}

]

}