BUSINESS INTELLIGENCE AND DATA ANALYTICS

Institute of Analytics

Open in Colab

The Perceptron

Implement the Perceptron using scikit-learn library

```
from sklearn.linear_model import Perceptron
import numpy as np

# Training data for AND gate
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([0, 0, 0, 1])

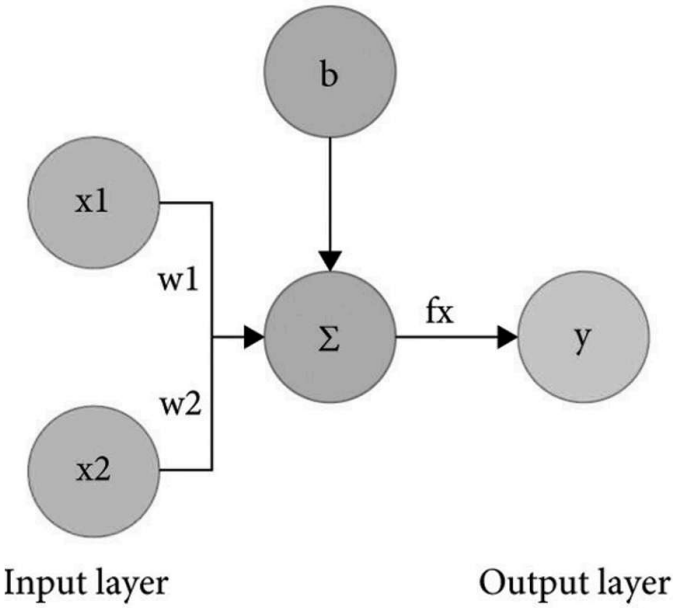
# Initialize and train Perceptron
model = Perceptron(max_iter=100, eta0=0.1, random_state=42)
model.fit(X, y)

# Results
print("Weights:", model.coef_)
print("Bias:", model.intercept_)
print("Predictions:", model.predict(X))
```

```
Weights: [[0.2 0.2]]
Bias: [-0.2]
Predictions: [0 0 0 1]
```

Note: In the scikit-learn Perceptron, the step function (also called the activation function) is a hard threshold function, and it's built-in.

```Python prediction = 1 if output >= 0 else 0




---

## The Mulit-Layer Perceptron - MLP

### Solving the XOR Problem with a Neural Network
```

This code demonstrates how to build and train a simple neural network from scratch using NumPy to learn the XOR logic gate.

```
from sklearn.neural_network import MLPClassifier
import numpy as np

# XOR input and output
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]])
y = np.array([0, 1, 1, 0])

# Define MLP with 1 hidden layer of 2 neurons (minimal config for XOR)
mlp = MLPClassifier(hidden_layer_sizes=(2,), activation='tanh',
                    solver='adam', learning_rate_init=0.01,
                    max_iter=10000, random_state=42)


# Train the model
mlp.fit(X, y)

# Make predictions
predictions = mlp.predict(X)

print("Predictions:\n", predictions)
print("\nWeights (input to hidden):\n", "[ w11 , w12 ]\n[ w21 , w22 ]\n", mlp.coefs_[0])
print("\nBias hidden:\n", mlp.intercepts_[0])
print("\nWeights (hidden to output):\n", mlp.coefs_[1])
print("\nBias output:\n", mlp.intercepts_[1])

```

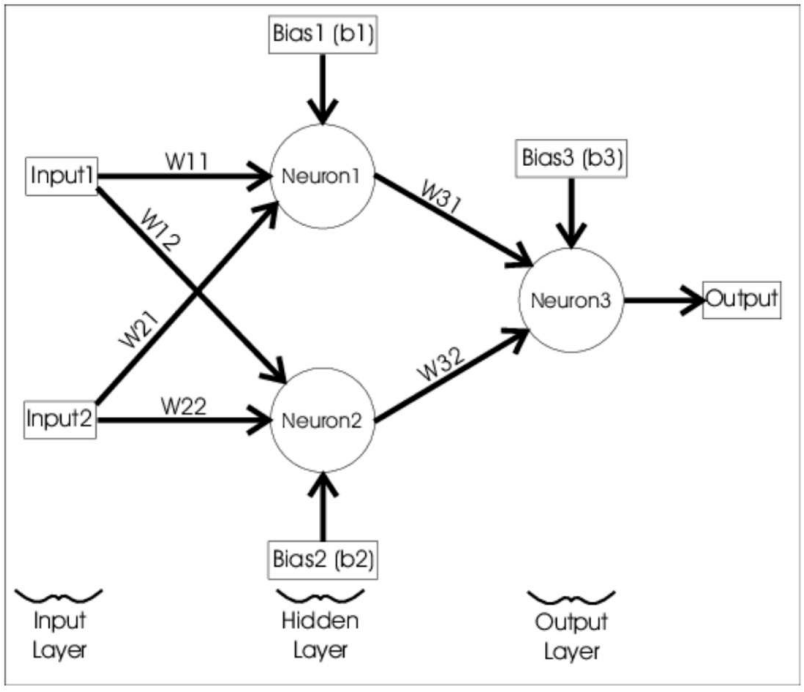
```
Predictions:
 [0 1 1 0]

Weights (input to hidden):
 [ w11 , w12 ]
[ w21 , w22 ]
 [[ 2.7144501   3.27401218]
 [-2.73418453 -3.17014048]]

Bias hidden:
 [ 1.21994174 -1.63451199]

Weights (hidden to output):
 [[-4.37775211]
 [ 4.46553876]]

Bias output:
 [3.61855675]
```



The Neural Network Model to solve the XOR Logic (from: https://stopsmokingaids.me/)

Building Word Embeddings from Scratch

We can build word embeddings from sratch using a corpus of our own and using gensim library to build Word2Vec representations.

Example 1 - Simple Tokenized Corpus

```
%pip install gensim
```

```
Collecting gensim
  Downloading gensim-4.3.3-cp310-cp310-win_amd64.whl (24.0 MB)
Collecting smart-open>=1.8.1
  Using cached smart_open-7.1.0-py3-none-any.whl (61 kB)
Collecting numpy<2.0,>=1.18.5
  Downloading numpy-1.26.4-cp310-cp310-win_amd64.whl (15.8 MB)
Collecting scipy<1.14.0,>=1.7.0
  Downloading scipy-1.13.1-cp310-cp310-win_amd64.whl (46.2 MB)
Collecting wrapt
  Using cached wrapt-1.17.2-cp310-cp310-win_amd64.whl (38 kB)
Installing collected packages: wrapt, numpy, smart-open, scipy, gensim
  Attempting uninstall: numpy
    Found existing installation: numpy 2.2.5
    Uninstalling numpy-2.2.5:
      Successfully uninstalled numpy-2.2.5
  Attempting uninstall: scipy
    Found existing installation: scipy 1.15.2
    Uninstalling scipy-1.15.2:
      Successfully uninstalled scipy-1.15.2
Successfully installed gensim-4.3.3 numpy-1.26.4 scipy-1.13.1 smart-open-7.1.0 wrapt-1.17.2
Note: you may need to restart the kernel to use updated packages.
  WARNING: The script f2py.exe is installed in 'c:\Users\me\AppData\Local\Programs\Python\Python310\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: You are using pip version 21.2.3; however, version 25.0.1 is available.
You should consider upgrading via the 'c:\Users\me\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' comm
```

```
from gensim.models import Word2Vec

# Sample corpus
sentences = [
    ['data', 'science', 'is', 'fun'],
    ['machine', 'learning', 'is', 'powerful'],
    ['data', 'and', 'learning', 'are', 'related']
]

# Train the model
model = Word2Vec(sentences, vector_size=50, window=2, min_count=1, workers=2)

# Access the embedding for a word
print("Vector for 'data':\n", model.wv['data'])

# Find similar words
print("Words similar to 'data':", model.wv.most_similar('data'))
```

```
Vector for 'data':
 [-0.01723938  0.00733148  0.01037977  0.01148388  0.01493384 -0.01233535
  0.00221123  0.01209456 -0.0056801  -0.01234705 -0.00082045 -0.0167379
 -0.01120002  0.01420908  0.00670508  0.01445134  0.01360049  0.01506148
 -0.00757831 -0.00112361  0.00469675 -0.00903806  0.01677746 -0.01971633
  0.01352928  0.00582883 -0.00986566  0.00879638 -0.00347915  0.01342277
  0.0199297  -0.00872489 -0.00119868 -0.01139127  0.00770164  0.00557325
  0.01378215  0.01220219  0.01907699  0.01854683  0.01579614 -0.01397901
 -0.01831173 -0.00071151 -0.00619968  0.01578863  0.01187715 -0.00309133
  0.00302193  0.00358008]
Words similar to 'data': [('are', 0.16563551127910614), ('fun', 0.13940520584583282), ('learning', 0.1267007291316986), ('powerful', 0.0
```

Example 2 - Simple Untokenized Corpus

The code below uses the gensim library to buidl word embeddings using Word2Vec models from scratch.
It uses a text corpus to learn word similarities.

```
%pip install nltk

import nltk
nltk.download('punkt_tab')
```

```
Requirement already satisfied: nltk in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from nltk) (8.1.8)
Requirement already satisfied: regex>=2021.8.3 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from nltk) (202
Requirement already satisfied: tqdm in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from nltk) (4.67.1)
Requirement already satisfied: joblib in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: colorama in c:\users\me\appdata\roaming\python\python310\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
WARNING: You are using pip version 21.2.3; however, version 25.0.1 is available.
You should consider upgrading via the 'c:\Users\me\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' comm
[nltk_data] Downloading package punkt to
[nltk_data]     C:/Users/me/AppData/Roaming/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

```
import gensim
from gensim.models import Word2Vec
from nltk.tokenize import word_tokenize

# Sample corpus
sentences = [
    "Large language models are transforming business applications",
    "Natural language processing helps computers understand human language",
    "Word embeddings capture semantic relationships between words",
    "Neural networks learn distributed representations of words",
    "Businesses use language models for various applications",
    "Customer service can be improved with language technology",
    "Modern language models require significant computing resources",
    "Language models can generate human-like text for businesses"
]

# Tokenize the sentences
tokenized_sentences = [word_tokenize(sentence.lower()) for sentence in sentences]

# Train Word2Vec model
model = Word2Vec(
    sentences=tokenized_sentences,
    vector_size=100,    # Embedding dimension
    window=5,           # Context window size
    min_count=1,        # Minimum word frequency
    workers=4           # Number of threads
)

# Save the model
model.save("word2vec.model")

# Find the most similar words to "language"
similar_words = model.wv.most_similar("language", topn=5)
print("Words most similar to 'language':")
for word, similarity in similar_words:
    print(f"{word}: {similarity:.4f}")

# Vector for a specific word
word_vector = model.wv["business"]
print(f"\nVector for 'business' (first 10 dimensions):\n{word_vector[:10]}")

# Word analogies
analogy_result = model.wv.most_similar(
    positive=["business", "language"],
    negative=["models"],
    topn=3
)
print("\nAnalogy results:")
for word, similarity in analogy_result:
    print(f"{word}: {similarity:.4f}")
```

```
Words most similar to 'language':
natural: 0.2196
between: 0.2167
resources: 0.1955
distributed: 0.1696
significant: 0.1522

Vector for 'business' (first 10 dimensions):
```

```
[ 0.00816812 -0.00444303  0.00898543  0.00825366 -0.00443522  0.00030311
  0.00427449 -0.00392632 -0.00555997 -0.00651232]

Analogy results:
neural: 0.2595
natural: 0.2004
resources: 0.1899
```

Word Similarities Examples

Use Fake Embeddings

```
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

# Fake word vectors (3D for simplicity)
word_vectors = {
  "king": np.array([0.8, 0.65, 0.1]),
   "queen": np.array([0.78, 0.66, 0.12]),
   "man": np.array([0.9, 0.1, 0.1]),
   "woman": np.array([0.88, 0.12, 0.12]),
  "apple": np.array([0.1, 0.8, 0.9]),
}
def similarity(w1, w2):
    return cosine_similarity([word_vectors[w1]], [word_vectors[w2]])[0][0]

print("Similarity(king, queen):", similarity("king", "queen"))
print("Similarity(man, woman):", similarity("man", "woman"))
print("Similarity(king, apple):", similarity("king", "apple"))
```

```
Similarity(king, queen): 0.9995995265529728
Similarity(man, woman): 0.999399810286
Similarity(king, apple): 0.5514092058274782
```

Use Real Embeddings - Gensim library

```
import gensim.downloader as api
from gensim.models import Word2Vec

# Load pre-trained Word2Vec model
word2vec_model = api.load("word2vec-google-news-300")

# Find similar words
similar_words = word2vec_model.most_similar('computer', topn=5)
print("Words similar to 'computer':", similar_words)

# Word analogies
result = word2vec_model.most_similar(positive=['woman', 'king'], negative=['man'], topn=1)
print("king - man + woman =", result)

# Train your own Word2Vec model
sentences = [["cat", "say", "meow"], ["dog", "say", "woof"]]
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)

# Get vector for a word
cat_vector = model.wv['cat']
print("Vector for 'cat':", cat_vector[:5])  # Show first 5 dimensions
```

```
Words similar to 'computer': [('computers', 0.7979382276535034), ('laptop', 0.6640493869781494), ('laptop_computer', 0.6548869013786316)
king - man + woman = [('queen', 0.7118191123008728)]
Vector for 'cat': [-0.00713902  0.00124103 -0.00717672 -0.00224462  0.0037193 ]
```

Use Real Embeddings - Spacy library

Download spacy and the required libraries

```
%pip install spacy
!python -m spacy download en_core_web_md
```

```
Requirement already satisfied: spacy in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (3.8.5)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in c:\users\me\appdata\local\programs\python\python310\lib\site-pack
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy)
Requirement already satisfied: packaging>=20.0 in c:\users\me\appdata\roaming\python\python310\site-packages (from spacy) (24.2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (frc
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from sp
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spac
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (frc
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from sp
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy)
Requirement already satisfied: jinja2 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy) (3.1.6)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy)
Requirement already satisfied: setuptools in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy) (57.4.0)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy
Requirement already satisfied: numpy>=1.19.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from spacy) (2.2.
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from sp
Requirement already satisfied: language-data>=1.2 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from langcod
Requirement already satisfied: marisa-trie>=1.1.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from languag
Requirement already satisfied: pydantic-core==2.33.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from pyda
Requirement already satisfied: typing-extensions>=4.12.2 in c:\users\me\appdata\roaming\python\python310\site-packages (from pydantic!=1
Requirement already satisfied: typing-inspection>=0.4.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from p
Requirement already satisfied: annotated-types>=0.6.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from pyd
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from r
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from request
Requirement already satisfied: idna<4,>=2.5 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from requests<3.0.
Requirement already satisfied: certifi>=2017.4.17 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from request
Requirement already satisfied: blis<1.4.0,>=1.3.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from thinc<8
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from t
Requirement already satisfied: colorama in c:\users\me\appdata\roaming\python\python310\site-packages (from tqdm<5.0.0,>=4.38.0->spacy)
Requirement already satisfied: click>=8.0.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from typer<1.0.0,>
Requirement already satisfied: shellingham>=1.3.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from typer<1
Requirement already satisfied: rich>=10.11.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from typer<1.0.0,
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\me\appdata\roaming\python\python310\site-packages (from rich>=10.11.0
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from rich
Requirement already satisfied: mdurl~=0.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from markdown-it-py>
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from w
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from
Requirement already satisfied: wrapt in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from smart-open<8.0.0,>=5
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\me\appdata\local\programs\python\python310\lib\site-packages (from jinja2->sp
Note: you may need to restart the kernel to use updated packages.
WARNING: You are using pip version 21.2.3; however, version 25.0.1 is available.
You should consider upgrading via the 'c:\Users\me\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' comm
Collecting en-core-web-md==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_md-3.8.0/en_core_web_md-3.8.0-py3-none-any.whl (33
Installing collected packages: en-core-web-md
Successfully installed en-core-web-md-3.8.0
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_md')
WARNING: You are using pip version 21.2.3; however, version 25.0.1 is available.
You should consider upgrading via the 'c:\Users\me\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' comm
```

```
import spacy
nlp = spacy.load("en_core_web_md")

word1 = nlp("king")
word2 = nlp("queen")
print("Similarity:", word1.similarity(word2))
```

```
Similarity: 0.38253095611315674
```