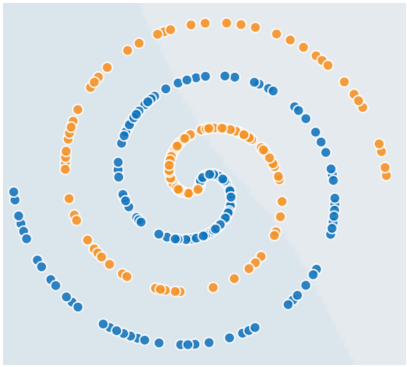
imal - Assignment 3

Introductie

In deze assignment gaan we een multi layer perceptron (MLP) trainen en testen op de spiraal dataset. Deze ben je eerder tegengekomen tijdens het werkcollege. Toen hebben we de playground gebruikt van Google's tensorflow:

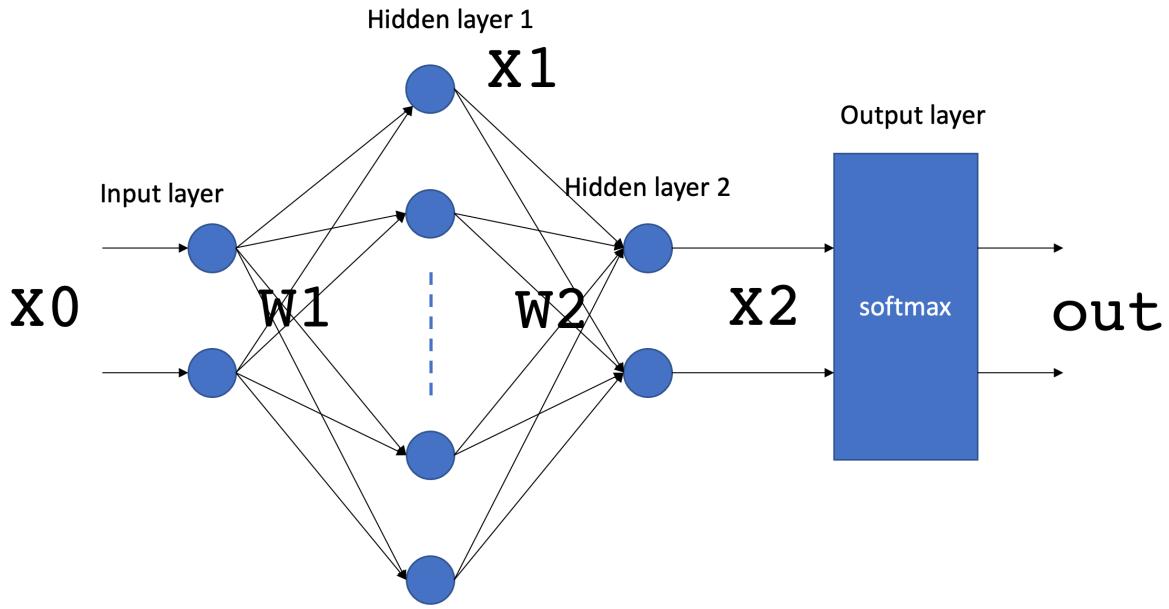


Figuur 1. De spiral dataset

https://playground.tensorflow.org

We gaan dit nu zelf namaken in ruwe python code. Dit betekent dat we alleen gebruiken zullen maken van de packages numpy, pandas en matplotlib. Het gebruik van andere packages is niet toegestaan (geen tensorflow). Het notebook `assignment3.ipynb` met ruwe code gaan we verder afmaken. Verder hebben we de dataset `spiral.csv`.

We gaan het volgende (zie figuur hieronder) neurale netwerk namaken. Een deel van dit neurale netwerk is al voor je geïmplementeerd in het notebook met de functies `trainMLP()` en `testMLP()`. Dit hoef je dus niet zelf te doen.



Figuur 2. De multilayer perceptron (MLP)

Dit neurale netwerk heeft een input layer, twee hidden layers en een output layer (softmax activatie functie). De activatie functie voor de hidden layers is de ReLU. De input X0 zijn de

twee dimensies van de spiraal dataset. Uiteindelijk wordt de input naar de volgende layer gestuurd met de weegfactoren W1 en een bias vector. Het resultaat X1 wordt berekend met de formule

```
X1 = ReLU( W1*X0 + B1)
```

En de tweede layer met

```
X2 = ReLU( W2*X1 + B2)
```

De hidden layer 1 heeft een aantal neuronen die straks nader moeten worden bepaald.

De hoofdfunctie `__main__` is ook al voor je geschreven en deze `__main__` roept alle functies aan die jij nog moet gaan implementen. De volgende functies moet je implementeren voor dit assignment:

- `leesspiraal()`
- `berekenaccuracy()`
- `normaliseerspiraldataset()`
- `getX()`
- `getY()`
- `getLabels()`
- `traintestsplit()`

Verder moeten in de `__main__` een aantal parameters bepaald worden namelijk:

- `epoch`
- `learn rate (lr)`
- `aantallayers (hidden layer 1)`

Opdrachten

<u>Opdracht 1: inlezen dataset</u>

De functie `leesspiraal()` moet je verder afmaken. De input van deze functie is het relatieve pad naar het bestand ('spiral.csv'). De output die de functie teruggeeft is een dataframe met de dimensies (900,3). In de `__main__` wordt deze als volgt aangeroepen:

```
df = leesspiraal('spiral.csv')
```

```
input: <class 'str'>
output: <class 'pandas.core.frame.DataFrame'>
output shape: (900, 3)
```

Opdracht 2: Bereken de accuracy

De functie `berekenaccuracy()` moet je verder afmaken. De input argumenten zijn twee numerieke waarden (integers) *correct* en *fout*. Met deze twee waarden bereken je de accuracy en deze accuracy geef je terug als output van de functie. In de `__main__` wordt deze als volgt aangeroepen:

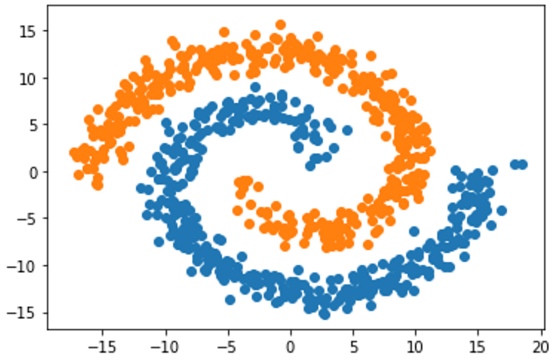
```
accuracy = berekenaccuracy(correct, fout)
```

```
input types: <class 'int'>
output type: <class 'float'>
```

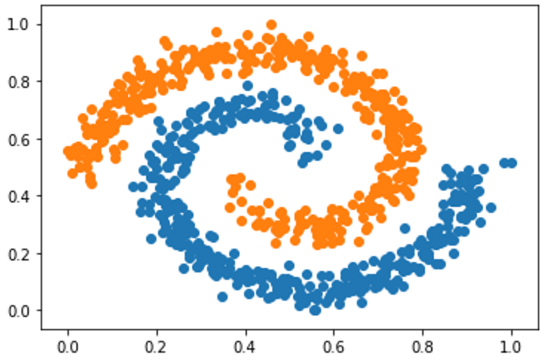
Opdracht 3: normaliseren van de data

De functie `normaliseerspiraldataset()` moet je verder afmaken. Normaliseren is een ander woord voor scaler of scaling (jullie zijn bekend met de StandardScaler). De input van deze functie verwacht een dataframe die eerder is ingeladen (spiraal). Je moet de kolommen X en Y normaliseren (scalen) tussen een waarden van 0 en 1. Het resultaat moet zijn zoals in onderstaande afbeelding.

Originele dataset



Genormaliseerde dataset



Normaliseren kun je als volgt doen:

1. Trek de minimale waarden in de dataset af van de originele data (hint: gebruik `np.min` functie).
2. Deel daarna het resultaat uit stap 1 door de maximale waarden uit stap 1. (hint: gebruik `np.max` functie).

Het resultaat geef je terug (output van de functie). Het resultaat is een dataframe met dezelfde dimensies als de input. In de `__main__` wordt deze als volgt aangeroepen:

```
df = normaliseerspiraldataset(df)
```

```
input type: <class 'pandas.core.frame.DataFrame'>
output type: <class 'pandas.core.frame.DataFrame'>
input shape: (900, 3)
output shape: (900, 3)
```

Opdracht 4: Selecteer de individuele kolommen uit de dataset

De functie `getX()` moet je verder afmaken. Deze functie heeft als input argument het originele dataframe. De output is de 'X' kolom uit het originele dataframe.

Doe ditzelfde voor de functies `getY()` en de functie `getLabel()`. Logischerwijs zie je aan de naamgeving van de functie welke kolom je moet teruggeven. In de `__main__` worden deze functies als volgt aangeroepen:

```
x_train = getX(train)
y_train = getY(train)
labels_train = getLabels(train)
```

```
input type: <class 'pandas.core.frame.DataFrame'>
output type: <class 'pandas.core.series.Series'>
input shape: (800, 3)
output shape: (800,)
```

Opdracht 5: Split de originele dataset in een train set en een test set

De functie `traintestsplit()` moet je verder afmaken. Deze functie heeft als input argument de originele dataset (spiral). De functie split de dataset op in een train gedeelte en een test gedeelte. Maak hiervoor gebruik van de functie `iloc`. En gebruik het onderstaande code voorbeeld van de volgende website:
https://sparkbyexamples.com/pandas/how-to-split-pandas-dataframe/
Het train gedeelte heeft uiteindelijk de dimensies (800,3) en het test gedeelte (100,3). De functie geeft een tuple terug met de train en test set. De functie word hier aangeroepen in de `__main__`.

```
train, test = traintestsplit(df)
```

```
input shape: (900, 3)
output shapes: (800, 3), (100, 3)
```

Opdracht 6: De hele code runnen en een goed resultaat behalen

Als alle functies werken (opdracht 1 t/m 5) is het tijd om het model te trainen en kijken of we een hoge accuracy kunnen bereiken. Voor het model moeten we een aantal parameters opgeven. Dit zijn:

- Aantal epochs
- Learning rate
- Aantal neuronen in de hidden layer.

In de `__main__` kun je deze parameters aan het begin opgeven. Het doel is om een accuracy te behalen van **hoger dan 0.95** op de test-data. Het script genereert een aantal files die je moet uploaden in Codegrade. Dit zijn de weegfactoren en de bias termen:

```
W1.npy
W2.npy
B1.npy
B2.npy
```

Oplevering in Codegrade

Je notebook met de naam assignment3.ipynb en de files uit opdracht 6.