

NNW-Übung 3

1 Mehrere Klassen

Ziel dieser Aufgabe ist, mehrere Klassen mittels eines Single-Layer Networks zu trennen. Als Beispieldaten sollen wieder die Iris-Daten dienen, wobei jetzt alle drei Iris Arten verwendet werden.

- a) Für die Konvertierung der mit der Klassennummer kodierten Zieldaten T (also z.B. Klasse 2) in die one-hot-Codierung für ein Netz (also im Beispiel $[0; 1; 0]$) können Sie die folgende Funktion zu Ihrer Klasse `SLN` hinzufügen:

```
def onehot(self,T):
    e=np.identity(self._W.shape[0])
    return e[:,T.astype(int)]
```

In der Funktion wird eine Einheitsmatrix (`np.identity(c)`) indiziert mit dem colon-Operator (`:`) und dem ursprünglichen Vektor der Zieldaten, der die Nummer der Klasse (Iris-Art) enthält.

Probieren Sie die Funktion zuerst für einen kleinen Zieldatenvektor `T=np.array([0,2,1,2])` aus und vollziehen Sie sie nach.

- b) Wie in der Vorlesung besprochen, soll die Zuordnung zu der Klasse erfolgen, deren zugehöriges Output-Neuron den größten Wert geliefert hat (siehe 03_Grundkonzepte, Folie 55¹ und 02_Aufgabenstellung, Folie 18 „one hot vectors“).

Beispiel: `np.array([[0.2],[0.6],[0.5]])` wird der Klasse 1 (Zählung ab 0) zugeordnet. Nützliche NumPy-Methode hierfür: `argmax`. Probieren Sie es aus. Ist es schon one-hot kodiert?

Ergänzen Sie nun die Klasse `SLN` wie folgt und vollziehen Sie den Code nach:

```
def __init__(self,dIn,cOut): # Konstruktor
    np.random.seed(42)
    self._b=np.zeros(cOut)[np.newaxis].T
    self._W=np.random.randn(cOut,dIn)/np.sqrt(dIn)
    if cOut==1:
        self.neuron=self.threshold
    else:
        self.neuron=self.thresholdMult
def netsum(self,X):
    return self._W.dot(X)+self._b
def threshold(self,X):
    return self.netsum(X)>=0
```

Implementieren Sie in der Klasse `SLN` eine Methode `thresholdMult(self,X)`, die die Netzwerksomme mit Hilfe von `netsum` berechnet und ein one-hot-kodiertes Ergebnis liefert.

- c) Trainieren Sie ein Single-Layer Network für die Unterscheidung aller drei Iris-Arten mit der Funktion `SlnDeltaTrain`.

Verwenden Sie dazu die Merkmale/Attribute 3 und 4!

Übernehmen Sie den Plot in die Auswertung und geben Sie zudem an: den von Ihnen gewählten Wert für η , die erreichte Fehlerrate und nach wievielen Iterationsschritten diese Fehlerrate *erstmalig* erreicht wurde.

Anmerkung zur Interpretation des Ergebnisses: Die Daten sind *nicht* linear separabel!

¹Die Foliennummer (oben links) ist gemeint, nicht die Seite im PDF!