

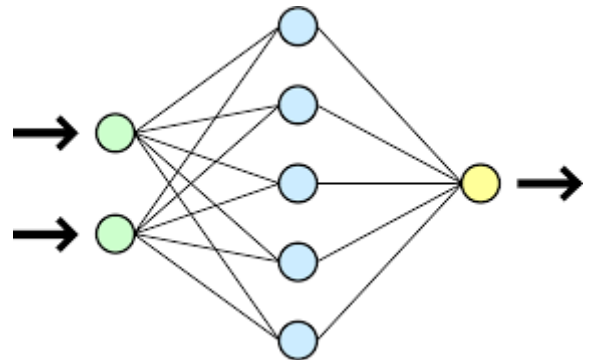
# Wie funktioniert unser KNN ?

KNN steht für künstliches neuronales Netz → Das heißt es lernt selbst auf gegebenen Trainingsdaten.

Unser KNN bekommt bereits vorverarbeitete Daten wie diese verarbeitet werden entnehmen sie bitte dem entsprechendem Plakat.

Unser KNN besteht aus vier Schichten:

Model: "model"

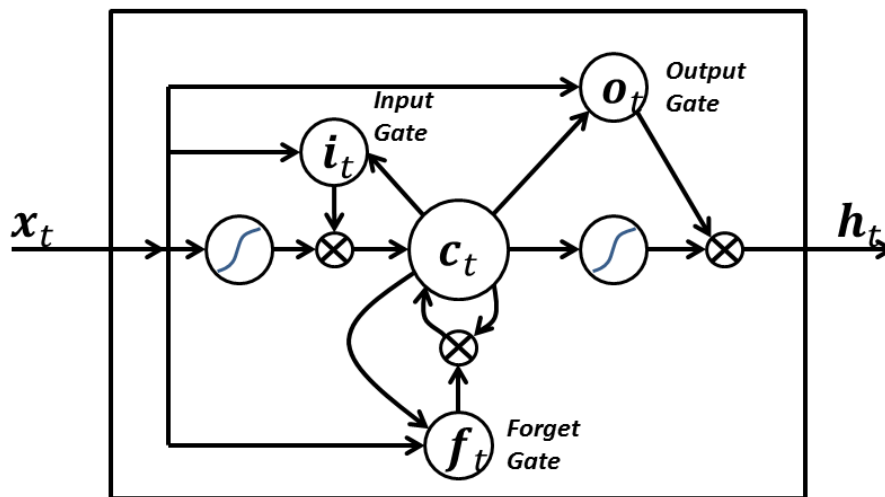


Layer (type)	Output Shape	Param #
lstm_input (InputLayer)	[(None, 40)]	0
lstm_embedding (Embedding)	(None, 40, 128)	7040
lstm (LSTM)	(None, 128)	131584
lstm_predictions (Dense)	(None, 40)	5160
Total params: 143,784		
Trainable params: 143,784		
Non-trainable params: 0		

Wobei die erste nur zu Eingabe da ist und nicht trainiert wird (sie hat keine trainierbaren Parameter)

- Embedding: Hier ändern wir die Dimension unseres Inputs dazu gibt es einige mathematische Algorithmen, die allerdings mehr Platz brauchen, als wir hier haben. Wir empfehlen Wikipedia für weitere Informationen. Nur so weit her werden unsere Muster die wir generiert haben in einen hochdimensionalen Raum projiziert. Es wird also irgend eine Matrix daraus generiert. Hier wird „trainiert wie Wörter zusammenhängen“ bzw. wie „nah“ sie sind.

- LSTM: Das ist der Wichtigste Teil unseres KNN. LSTM steht für Long Short Time Memory und ist besonders für Sprach und Text Verarbeitung geeignet da es sich Teile des bisherigen Inputs merken kann.



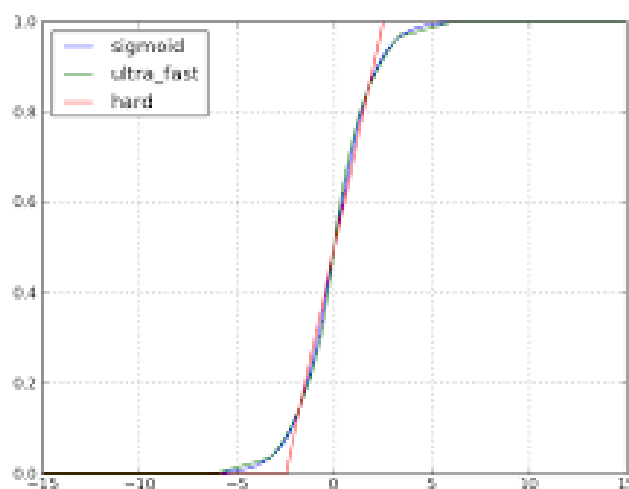
Gates greifen auf internen „Speicher“ zu

- Input-Gate: Wie viel des neuen Inputs wird aufgenommen?
- Forget-Gate: Wie viele Teile sollen aus dem „Speicher“ gelöscht werden
- Output-Gate: Verarbeitung des Inputs unter Verwendung des „Speichers“.

In den Gates sind Sigmoide Funktionen die das mit einer Zahl zwischen 1 und 0 entscheiden. Während des Trainings wird sowohl der „Speicher“ als auch die Gates und andere enthaltene Gewichte oder Aktivierungsfunktionen angepasst um bessere Ergebnisse zu produzieren.

#### 4. Sigmoide:

Hier haben wir ein ganz simples komplett verbundenes neuronales Netz, welches aus dem LSTM vorhersage eine Prozentpunkte Skala zwischen 0 und 1 macht dadurch wird das schlussendliche Punkte setzen erleichtert, wir nutzen hier eine harte Sigmoide um klarere Ergebnisse zu erzielen.



Die sieht etwa so wie die rote Linie aus

Später wollen wir diese Funktion durch eine unit step function ersetzen die so aussieht:

