

# TI2736-A

## Assignment 1: Artificial Neural Network

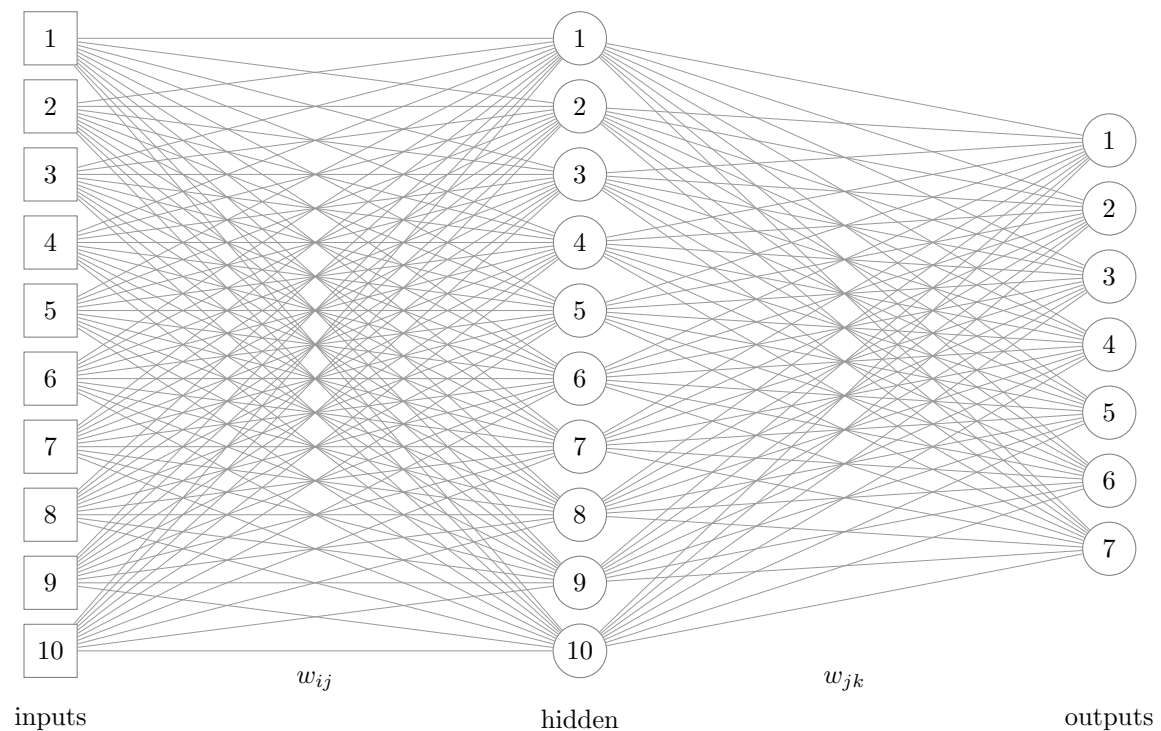
David Akkerman -  
Jan Pieter Waagmeester - 1222848

18 september 2014

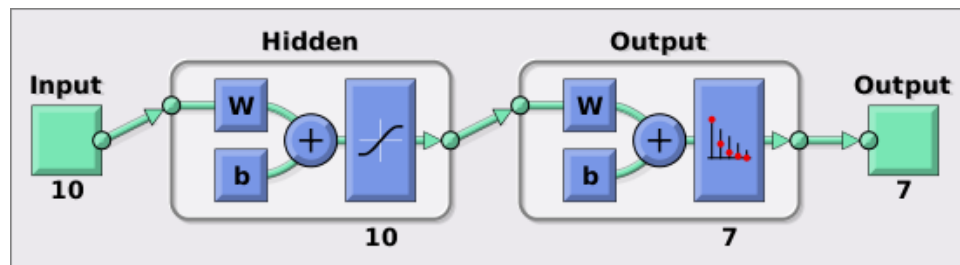
1. Omdat er 10 eigenschappen zijn hebben we 10 inputneurons nodig.
2. Voor de 7 verschillende classes hebben we 7 outputneurons nodig.
3. We beginnen met 10 hidden neurons.
4. We zullen de sigmoid-functie gebruiken:

$$Y^{sigmoid} = \left( \frac{1}{1 + e^{-x}} \right)$$

5. Ons netwerk kan voorgesteld worden als in het volgende plaatje. Daarbij kan het aantal hidden neurons uiteraard gevarieerd worden.



6. We gebruiken ongeveer 50% van de data voor training, 25% voor validatie en 25% als testset. De data lijkt aardig random gesorteerd te zijn, het is dus niet nodig om te zorgen dat er ongeveer even veel van elke class in elke set aanwezig is.
7. Door te kijken naar de *Mean square error* (MSE) op te trainingset, en op de validatieset.
- 8.
- 9.
- 10.
- 11.
- 12.
- 13.
- 14.
15. Met `nprtool` is het binnen een paar minuten mogelijk om een netwerk te maken wat er zo uit ziet:



16. De performance van dit netwerk is erg goed, en de training is ook erg snel klaar. Zelf met 10 hidden neurons worden succes-percentages van boven de 90% gehaald.  
 Tijdens het maken van het eigen neurale netwerk had ik het idee dat het gebruiken van de sigmoid-activatiefunctie wellicht voor de output-layer niet handig was. Dit netwerk gebruikt een softmax activatiefunctie in de output-layer.  
 De veel hogere performance is waarschijnlijk verklaarbaar door bijvoorbeeld implementatie in correcte matrix-algebra en het toevoegen van het aanpassen van de *learning rate*, het toevoegen van een momentum-term in de delta-rule (Negnevitsky, equation 6.17).