

# Neurala nätverk & Keras

en introduktion

# Introduktion

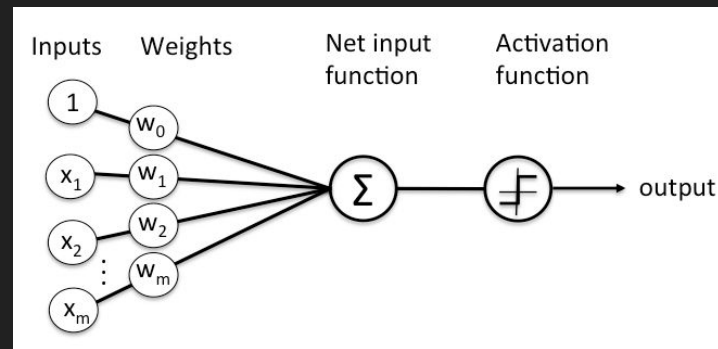
- Syfte
  - Sprida intresset för machine learning
  - Ge en grund för att börja arbeta med NN
  - Visa på enkelheten att bygga och använda neurala nätverk
- Att börja med:
  - Neural network basics med fokus på klassificering
  - En kort introduktion av Keras
  - Ett ganska trivialt exempel
- Om tid över:
  - Ett inte lika trivialt exempel

# Vad är ett neuralt nätverk?

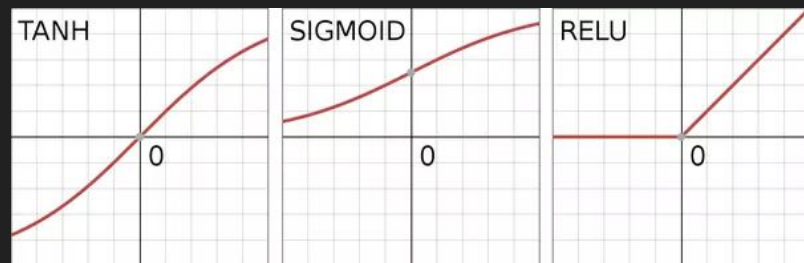
- En (kedja av) funktion(er)
  - $f: \mathbf{X}, \mathbf{W} \rightarrow \mathbf{Y}$
- Ett antal beräkningsenheter, ordnade i lager, kopplade med viktade länkar
- En metod för att approximera funktioner
- Ett verktyg för att modellera ett *klassificeringsproblem*
  - Binary: representerar bilden en katt?
  - Multiclass: representerar bilden en "hund", "katt", eller "mus"?
  - Diskret: "ja", "nej", "hund"
  - Probabilistiskt: sannolikhetsfördelning över klasserna

# Neurala nätverkets komponenter: nod

- En beräkningsenhet
- Realiserar en funktion
  - $y = a(\mathbf{x} \bullet \mathbf{w})$
  - Output motsvarar nodens “aktivering”
- Ett antal inputs ( $\mathbf{x} = x_1, \dots, x_n$ )
  - (Ofta låter man en input alltid vara = 1 (bias))
- Varje input har en vikt ( $\mathbf{w} = w_1, \dots, w_n$ )
- Input multipliceras vikt och summeras
- Aktiveringsfunktion  $a$ 
  - Icke-linjär funktion
  - Syfte: att tillgängliggöra icke-linjära mappning
  - ReLU, Sigmoid, tanh, m. fl.



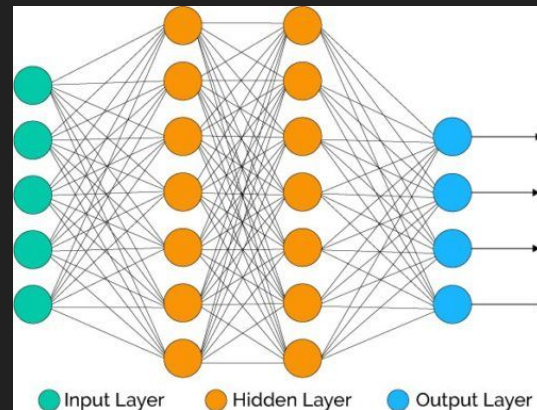
Bildkälla: deeplearning4j.org



Bildkälla: Danijar Hafner, quora.org

# Neurala nätverkets komponenter: lager

- Input layer
  - Behållare för input (feature-vektor)
  - Varje “nod” realiserar en identitetsfunktion, input = output
- Hidden layers
  - Feature-detektorer, mönsterigenkännare
  - Idé: noder ska aktiveras i relation till mönster i input
- Output layer
  - Omvandlar de dolda lagrens aktivering till en skala vi kan tolka
- Sammankopplade med viktade länkar
- I feedforward neural network gäller:
  - Data färdas från input-lagret, genom de dolda, till output-lagret
  - Output från ett lager blir input till det följande



Bildkälla: mtechprojects.org

# Neurala nätverkets komponenter: sammanfattning

- En samling sammankopplade beräkningsenheter (noder) ordnade i lager
- Mapper input till output, givet interna parametrar (vikter)
- Input ges till det första lagret
- Varje nod i hidden och output layers
  1. Summerar den viktade outputen från det föregående lagret
  2. Applicerar aktiveringsfunktion på summan
- Aktiveringen i det sista lagret utgör nätverkets output

# När är neurala nätverk lämpligt?

- NN är mer lämpligt:
  - när stor mängd (märkt) data finns tillgängligt
  - när vi har liten kännedom om vilka egenskaper som beskriver klasserna
  - när datat har kontinuerlig karaktär
- NN är mindre lämpligt:
  - när lärande inte är nödvändigt, t. ex. om problemet är linjärt separabelt
  - när kännedom om varför en viss egenskap leder till ett visst resultat är avgörande

# Hur löser vi klassificeringsproblem med NN?

1. Förbered data
2. Konstruera modellen (nätverket)
3. Fitting: träna modellen
4. Evaluera modellen



# 1. Förbered data

- Märkt data utgörs av sample-label-par
- Sample: ett stickprov, en observation, en feature-vektor, input
- Label: klassen ett sample tillhör, "hund", det önskade outputen
- Dela upp samples i två ömsesidigt uteslutande mängder
- Båda bör representera problemet väl
- Båda bör ha samma klassdistribution
- Training set
  - Samples att träna modellen med
- Test set
  - Samples att evaluera den tränade modellen på

## 2. Konstruera modellen (1/2)

- Struktur
  - Input-”noder”
    - Antalet ska matcha dimensionen på feature-vektorn
  - Output-noder
    - Antalet ska matcha antalet klasser (eller 1 om binärt)
  - Dolda lager och noder
    - Ingen gimmie
    - Trial and error
    - “Komplexare problem, fler noder”

## 2. Konstruera modellen (2/2)

- Activation function
  - Bestäms per lager
- Loss/cost function
  - Representerar kostnaden av felaktigheter i output
  - Ett mått på hur väl modellen fungerar i lärandet
  - Hur mycket skiljer sig svaret från sanningen?
  - Beräknas på nätverkets output och det rätta svaret
  - Ex: mean squared error (MSE)
- Optimizer method
  - Beskriver hur modellen uppdaterar sina inre parametrar (vikter)
  - Ex: stochastic gradient descent (SGD)

### 3. Träna modellen

- Syfte: lära modellen om koppling mellan feature och label
- Mål: minimera loss function
- Princip: beräkna det effektivaste sättet att minska loss function
- Steg (i grova drag):
  1. Mata nätverket med ett antal samples och beräkna medelkostnaden av samlad output
  2. Beräkna proportionen av varje vikts bidrag till kostnadsfunktionen (gradient)
    - Backpropagation – lärande-algoritmen
  3. Korrigera vikterna i gradientens negativa riktning
    - Stochastic gradient descent – “ta dig ner till botten av dalen”
    - Hur *mycket* man korrigerar vikterna styrs av *learning rate*

## 4. Evaluera modellen

- Syfte: att utvärdera hur väl modellen generaliserar problemet
- Mata nätverket med samples från test set
- Jämför resultaten med de önskade (labels)
- Hur mäter vi prestanda?
  - Accuracy
    - andelen korrekta klassificeringar bland alla klassificeringar
    - $TP + TN / (TP + TP + FP + FN)$
  - Precision & Recall
    - Precision: Positive Predictive Value,  $(TP / (TP + FP))$
    - Recall: True Positive Rate, Sensitivity  $(TP / (TP + FN))$
  - Med flera...

# Keras

Från [keras.io](https://keras.io):

- High-level neural network API
- Utvecklat med fokus på att tillgängliggöra snabb experimentation
- Stödjer olika typer av nätverk
- Stödjer körning på både CPU och GPU
- Användarvänligt
- Hög modularitet

# Keras – begrepp

- Model: representerar en modell av ett neural nätverk
  - Håller både struktur och saker som loss function, metrics
  - Sequential: modell för ett typiskt feedforward neural network
- Layer: representerar ett lager i ett nätverk
  - Dense: fully connected layer
- Batch: ett antal samples, processas oberoende och parallellt
- Epoch: en iteration av hela tränings-set:et

# Ett ganska trivialt exempel

Logisk OR-grind

Input	Önskad output
(0, 0)	0
(0, 1)	1
(1, 0)	1
(1, 1)	1

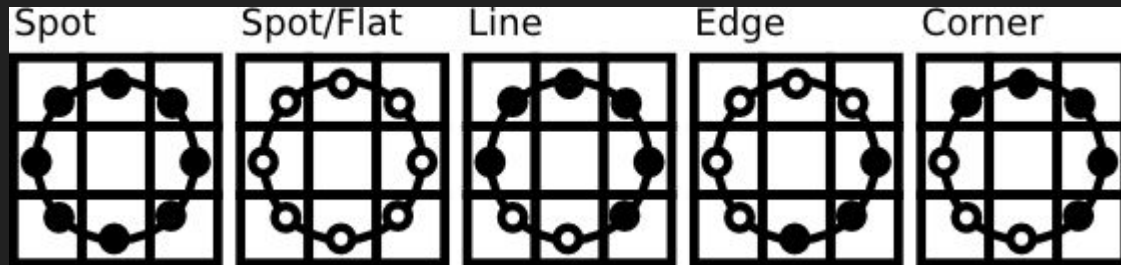


# Ett mindre trivialt exempel

Stad eller by?

Input	Önskad output
LBP(satellitbild över del av stad)	0
LBP(satellitbild över del av by)	1

LBP i en mening: frekvenser av olika intensitetsmönster i en bild



Bildkälla: [opencv.org](https://opencv.org)

# Vad vi förhoppningsvis lärt oss

- Olika synsätt på vad ett neuralt nätverk är
- Beståndsdelar i neurala nätverk
- Stegen i att lösa ett klassificeringsproblem med neurala nätverk
- Hur man gör NN i Keras