Neural Networks
Assignment 1

Aniruddha Pal, Mohammad Wasil, Wei-Chan Hsu

October 9, 2017

1 Summary of chapter 1

- A neural network is a massively parallel distributed processor made up of many information-processing units called neurons. It resembles the human brain in two respects:
  - Knowledge is acquired through learning.
  - The acquired knowledge is stored by interneuron connection strengths known as synaptic weights.
- A neuron consists of three basic elements:
  - A set of synapses or connecting links
  - An adder for summing the weighted input signals
  - An activation function for restricting on the output amplitude. It defines the output of the neuron in terms of the induced local field. There are three types:
    - Threshold function
    - Piecewise-linear function
    - Sigmoid function
- Neural networks can be represented as directed graphs using signal-flow graphs.
- Neural networks can be graphically represented in three ways:
  - Block diagram
  - Signal-flow graph (complete directed graph)
  - Architectural graph (partially complete directed graph)
- Feedback, allowing a network to be dynamic, plays a major role in recurrent neural networks.
- For a first-order infinite-duration impulse response (IIR) filter, the weight controls the dynamic behavior of the system: the output converges if $|w| < 1$.
- There are three classes of network architectures:
  - Single-layer feedforward networks
  - Multilayer feedforward netwroks
  - Recurrent networks

- A major task of a neural network is to learn and maintain a model of the world. Knowledge of the world can be either prior information or observations.
- Knowledge representation of the environment is defined by the synaptic weights and biases, which determine the performance of the network. There are four rules:
  - Similar inputs from similar classes produce similar representation inside the network.
  - Items to be categorized as separate classes should be given different representation in the network.
  - There should be numerous neurons involved in the representation of an important feature.
  - Prior information and invariances should be built into the design of neural networks.
- An AI system has three key components: representation, reasoning, and learning.

2 Models of a neuron

- **1.1**

1.1 An example of the logistic function is defined by

$$\varphi(v) = \frac{1}{1 + exp(-av)}$$

whose limiting values are 0 and 1. Show that the derivative of $\varphi(v)$ wrt. v is given by

$$\frac{d\varphi}{dv} = a\varphi(v)[1 - \varphi(v)]$$

what is the value of this derivative at the origin?

```
v,a = symbols('v a')
phi_v = 1/(1+exp(-a*v))
phi_vprime = diff(phi_v,v)
pprint(phi_vprime)
```

$$\frac{a \cdot e^{-a \cdot v}}{\left(1 + e^{-a \cdot v}\right)^2}$$

Value of this derivative at the origin is:

```
pprint(phi_vprime.subs(v,0))
```

$$\frac{a}{4}$$

- **1.3**

1.3 Yet another odd sigmoid function is the algebraic sigmoid:

$$\varphi(v) = \frac{v}{\sqrt{1 + v^2}}$$

whose limiting values are -1 and +1. Show that the derivative of $\varphi(v)$ wrt. v is given by

$$\frac{d\varphi}{dv} = \frac{\varphi^3(v)}{v^3}$$

what is the value of this derivative at the origin?

```
v,a = symbols('v a')
phi_v2 = v/sqrt(1+v*v)
phi_v2prime = phi_v2.diff(v)
pprint(phi_v2prime)
```

$$-\frac{v^2}{\left(v^2+1\right)^{3/2}} + \frac{1}{\sqrt{v^2+1}}$$

Value of this derivative at the origin is:

```
pprint(phi_v2prime.subs(v,0))
```

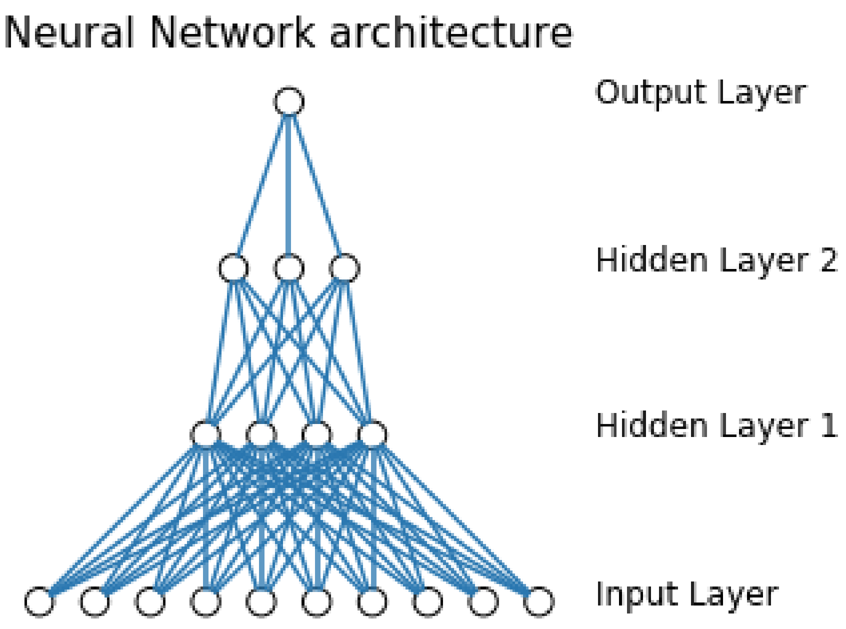
1

3 Network architectures

- **1.12**

1.12 A fully connedted feedforward network has 10 source nodes, 2 hidden layers, one with 4 neurons and the other with 3 neurons, and a single output neuron. Construct an architechtural graph of this network.

```
network = plotNN.DrawNN( [10,4,3,1] )
network.draw()
```



- **1.13**

1.13
a. Figure P1.13 shows the signal-flow graph of a 2-2-2-1 feedforward network. The function xi denotes a logistic function. Write the input-output mapping defined by this network.

```
x_1, x_2, a = symbols('x_1,x_2,a')
uk_1, uk_2, uk_3, uk_4, uk_5 = symbols('uk_1, uk_2, uk_3, uk_4, uk_5')
vk_1, vk_2, vk_3, vk_4, vk_5 = symbols('vk_1, vk_2, vk_3, vk_4, vk_5')
yk_1, yk_1, yk_2, yk_3, yk_4, yk_5 = symbols('yk_1, yk_1, yk_2, yk_3, yk_4, yk_5')

uk_1 = 5*x_1 + 1*x_2
uk_2 = 2*x_1 - 3*x_2

vk_1 = uk_1 #Normally + bias
vk_2 = uk_2

#Calculate the output using logistic function
yk_1 = 1/(1 + exp(-a * vk_1))
yk_2 = 1/(1 + exp(-a * vk_2))

uk_3 = 3*yk_1 + 4*yk_1
uk_4 = -1*yk_2 + 6*yk_2

vk_3 = uk_3
vk_4 = uk_4

yk_3 = 1/(1 + exp(-a * vk_3))
yk_4 = 1/(1 + exp(-a * vk_4))

uk_5 = -2*yk_3 + 1*yk_4

vk_5 = uk_5

yk_5 = 1/(1 + exp(-a * vk_5))
```

```
pprint(yk_5)
```

$$\frac{1}{1 + e^{-a\cdot\left(-\frac{2}{1 + e^{\frac{-7\cdot a}{1 + e^{-a\cdot(5\cdot x_1 + x_2)}}}} + \frac{1}{1 + e^{\frac{-5\cdot a}{1 + e^{-a\cdot(2\cdot x_1 - 3\cdot x_2)}}}}\right)}}$$

b. Suppose that the output neuron in the signal-flow graph of Fig. P1.13 operates in its linear region. Write the input-output mapping defined by this network.

```
x_1, x_2, a = symbols('x_1,x_2,a')
uk_1, uk_2 = symbols('uk_1, uk_2')
vk_1, vk_2, vk_3, vk_4 = symbols('vk_1, vk_2, vk_3, vk_4')
yk = symbols('yk')

uk_1 = 5*x_1 + 1*x_2
uk_2 = 2*x_1 - 3*x_2

vk_1 = uk_1
vk_2 = uk_2

vk_3 = 3*uk_1 + -1*uk_2

vk_4 = 4*uk_1 + 6*uk_2

yk = -2*vk_3 + 1*vk_4
```

The output of this network is

```
yk
```

$6x_1 - 26x_2$

4 Knowledge representation

- **1.21**

1.21
Let x be an input vector, and $s(\alpha, x)$ be a transformation operator acting on x and depending on some parameter alpha. The operator $s(\alpha, x)$ satisfies two requirements:

- $s(0, x) = x$
- $s(\alpha, x)$ is differentiable wrt. alpha.

The tangent vector is defined by the partial derivatives $\partial s(\alpha, x)/\partial a$ (Simard et al., 1992).
Suppose that x represents an image, and alpha is a rotation parameter. How would you compute the tanget vector for the case when alpha is small? The tangent vector is locally invariant wrt. rotation of the original image; why?

We can apply taylor series for very small values of α to get $s(\alpha, \mathbf{x})$

$$s(\alpha, \mathbf{x}) = s(0, \mathbf{x}) + \alpha \frac{\partial s(\alpha, \mathbf{x})}{\partial \alpha}$$

Where $s(\alpha, \mathbf{x})$ is tramsformation parameter acting on \mathbf{x} depending on some parameter α.
Since tangent vector v is defined by partial derivative $\partial s(\alpha, \mathbf{x})/\partial \mathbf{x}$. We can write

$$s(\alpha, \mathbf{x}) = s(0, \mathbf{x}) + \alpha v$$

Therefore

$$\begin{aligned} v &= \frac{s(\alpha, \mathbf{x}) - s(0, \mathbf{x})}{\alpha} \\ &\approx \frac{s(0, \mathbf{x}) - s(0, \mathbf{x})}{\alpha} = 0 \end{aligned}$$

The tangent vector is locally invariant with respect to rotation of the image of the original image $\mathbf{x}$, because the tangent vector becomes 0 for small α and is independent of $\mathbf{x}$.