Natural Language Processing 1

Lecture 6: Distributional semantics: generalisation and word embeddings

Katia Shutova

ILLC
University of Amsterdam

15 November 2018

Experimental corpus

- Dump of entire **English Wikipedia**, parsed with the English Resource Grammar producing dependencies.
- Dependencies include:
  - **For nouns**: head verbs (+ any other argument of the verb), modifying adjectives, head prepositions (+ any other argument of the preposition).
    cat: *chase_v+mouse_n, black_a, of_p+neighbour_n*
  - **For verbs**: arguments (NPs and PPs), adverbial modifiers.
    eat: *cat_n+mouse_n, in_p+kitchen_n, fast_a*
  - **For adjectives**: modified nouns; head prepositions (+ any other argument of the preposition)
    black: *cat_n, at_p+dog_n*

System description

- Semantic space: top 100,000 contexts.
- Weighting: pointwise mutual information (PMI).

An example noun

- *language*:

0.54::other+than_p+English_n
0.53::English_n+as_p
0.52::English_n+be_v
0.49::english_a
0.48::and_c+literature_n
0.48::people_n+speak_v
0.47::French_n+be_v
0.46::Spanish_n+be_v
0.46::and_c+dialects_n
0.45::grammar_n+of_p
0.45::foreign_a
0.45::germanic_a
0.44::German_n+be_v
0.44::of_p+instruction_n
0.44::speaker_n+of_p
0.42::pron_rel_+speak_v
0.42::colon_v+English_n
0.42::be_v+English_n
0.42::language_n+be_v
0.42::and_c+culture_n
0.41::arabic_a
0.41::dialects_n+of_p
0.40::percent_n+speak_v
0.39::spanish_a
0.39::welsh_a
0.39::tonal_a

An example adjective

- *academic*:

0.52::Decathlon_n
0.51::excellence_n
0.45::dishonesty_n
0.45::rigor_n
0.43::achievement_n
0.42::discipline_n
0.40::vice_president_n+for_p
0.39::institution_n
0.39::credentials_n
0.38::journal_n
0.37::journal_n+be_v
0.37::vocational_a
0.37::student_n+achieve_v
0.36::athletic_a
0.36::reputation_n+for_p
0.35::regalia_n
0.35::program_n
0.35::freedom_n
0.35::student_n+with_p
0.35::curriculum_n
0.34::standard_n
0.34::at_p+institution_n
0.34::career_n
0.34::Career_n
0.33::dress_n
0.33::scholarship_n
0.33::prepare_v+student_n
0.33::qualification_n

Corpus choice

- As much data as possible?
  - British National Corpus (BNC): 100 m words
  - Wikipedia: 897 m words
  - UKWac: 2 bn words
  - ...
- In general preferable, *but*:
  - More data is not necessarily the data you want.
  - More data is not necessarily realistic from a psycholinguistic point of view. We perhaps encounter 50,000 words a day. BNC = 5 years' text exposure.

Data sparsity

- Distribution for *unicycle*, as obtained from Wikipedia.

0.45::motorized_a
0.40::pron_rel_+ride_v
0.24::for_p+entertainment_n
0.24::half_n+be_v
0.24::unwieldy_a
0.23::earn_v+point_n
0.22::pron_rel_+crash_v
0.19::man_n+on_p
0.19::on_p+stage_n
0.19::position_n+on_p

0.17::slip_v
0.16::and_c+1_n
0.16::autonomous_a
0.16::balance_v
0.13::tall_a
0.12::fast_a
0.11::red_a
0.07::come_v
0.06::high_a

Polysemy

- Distribution for *pot*, as obtained from Wikipedia.

0.57::melt_v
0.44::pron_rel_+smoke_v
0.43::of_p+gold_n
0.41::porous_a
0.40::of_p+tea_n
0.39::player_n+win_v
0.39::money_n+in_p
0.38::of_p+coffee_n
0.33::amount_n+in_p
0.33::ceramic_a
0.33::hot_a

0.32::boil_v
0.31::bowl_n+and_c
0.31::ingredient_n+in_p
0.30::plant_n+in_p
0.30::simmer_v
0.29::pot_n+and_c
0.28::bottom_n+of_p
0.28::of_p+flower_n
0.28::of_p+water_n
0.28::food_n+in_p

Polysemy

- Some researchers incorporate word sense disambiguation techniques.
- But most assume a single space for each word: can perhaps think of subspaces corresponding to senses.
- Graded rather than absolute notion of polysemy.

Idiomatic expressions

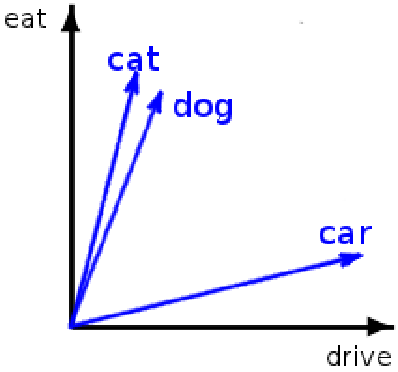
- Distribution for *time*, as obtained from Wikipedia.

0.46::of_p+death_n
0.45::same_a
0.45::1_n+at_p(temp)
0.45::Nick_n+of_p
0.42::spare_a
0.42::playoffs_n+for_p
0.42::of_p+retirement_n
0.41::of_p+release_n
0.40::pron_rel_+spend_v
0.39::sand_n+of_p
0.39::pron_rel_+waste_v

0.38::place_n+around_p
0.38::of_p+arrival_n
0.38::of_p+completion_n
0.37::after_p+time_n
0.37::of_p+arrest_n
0.37::country_n+at_p
0.37::age_n+at_p
0.37::space_n+and_c
0.37::in_p+career_n
0.37::world_n+at_p

Calculating similarity in a distributional space

- Distributions are vectors, so distance can be calculated.



Measuring similarity

- Cosine:

$$cos(\theta) = \frac{\sum v1_k * v2_k}{\sqrt{\sum v1_k^2} * \sqrt{\sum v2_k^2}} \quad (1)$$

- The cosine measure calculates the angle between two vectors and is therefore length-independent. This is important, as frequent words have longer vectors than less frequent ones.
- Other measures include Jaccard, Euclidean distance etc.

The scale of similarity: some examples

house – building 0.43
gem – jewel 0.31
capitalism – communism 0.29
motorcycle – bike 0.29
test – exam 0.27
school – student 0.25
singer – academic 0.17
horse – farm 0.13
man –accident 0.09
tree – auction 0.02
cat –county 0.007

Words most similar to *cat*

as chosen from the 5000 most frequent nouns in Wikipedia.

1 cat	0.29 human	0.25 woman	0.22 monster
0.45 dog	0.29 goat	0.25 fish	0.22 people
0.36 animal	0.28 snake	0.24 squirrel	0.22 tiger
0.34 rat	0.28 bear	0.24 dragon	0.22 mammal
0.33 rabbit	0.28 man	0.24 frog	0.21 bat
0.33 pig	0.28 cow	0.23 baby	0.21 duck
0.31 monkey	0.26 fox	0.23 child	0.21 cattle
0.31 bird	0.26 girl	0.23 lion	0.21 dinosaur
0.30 horse	0.26 sheep	0.23 person	0.21 character
0.29 mouse	0.26 boy	0.23 pet	0.21 kid
0.29 wolf	0.26 elephant	0.23 lizard	0.21 turtle
0.29 creature	0.25 deer	0.23 chicken	0.20 robot

But what is similarity?

- In distributional semantics, very broad notion: synonyms, near-synonyms, hyponyms, taxonomical siblings, antonyms, etc.
- Correlates with a psychological reality.
- Test via correlation with human judgments on a test set:
  - Miller & Charles (1991)
  - WordSim
  - MEN
  - SimLex

Miller & Charles 1991

3.92 automobile-car	3.05 bird-cock	0.84 forest-graveyard
3.84 journey-voyage	2.97 bird-crane	0.55 monk-slave
3.84 gem-jewel	2.95 implement-tool	0.42 lad-wizard
3.76 boy-lad	2.82 brother-monk	0.42 coast-forest
3.7 coast-shore	1.68 crane-implement	0.13 cord-smile
3.61 asylum-madhouse	1.66 brother-lad	0.11 glass-magician
3.5 magician-wizard	1.16 car-journey	0.08 rooster-voyage
3.42 midday-noon	1.1 monk-oracle	0.08 noon-string
3.11 furnace-stove	0.89 food-rooster	
3.08 food-fruit	0.87 coast-hill	

- Distributional systems, reported correlations 0.8 or more.

TOEFL synonym test

Test of English as a Foreign Language: task is to find the best match to a word:

Prompt:		levied
Choices:	(a)	imposed
	(b)	believed
	(c)	requested
	(d)	correlated
Solution:	(a)	imposed

- Non-native English speakers applying to college in US reported to average 65%
- Best corpus-based results are 100%

Distributional methods are a usage representation

- Distributions are a good conceptual representation if you believe that 'the meaning of a word is given by its usage'.
- Corpus-dependent, culture-dependent, register-dependent.
  Example: similarity between *policeman* and *cop*: 0.23

Distribution for *policeman*

policeman

0.59::ball_n+poss_rel
0.48::and_c+civilian_n
0.42::soldier_n+and_c
0.41::and_c+soldier_n
0.38::secret_a
0.37::people_n+include_v
0.37::corrupt_a
0.36::uniformed_a
0.35::uniform_n+poss_rel
0.35::civilian_n+and_c
0.31::iraqi_a
0.31::lot_n+poss_rel
0.31::chechen_a
0.30::laugh_v
0.29::and_c+criminal_n
0.28::incompetent_a
0.28::pron_rel_+shoot_v
0.28::hat_n+poss_rel
0.28::terrorist_n+and_c
0.27::and_c+crowd_n
0.27::military_a
0.27::helmet_n+poss_rel
0.27::father_n+be_v
0.26::on_p+duty_n
0.25::salary_n+poss_rel
0.25::on_p+horseback_n
0.25::armed_a
0.24::and_c+nurse_n
0.24::job_n+as_p
0.24::open_v+fire_n

Distribution for *cop*

cop

0.45::crooked_a
0.45::corrupt_a
0.44::maniac_a
0.38::dirty_a
0.37::honest_a
0.36::uniformed_a
0.35::tough_a
0.33::pron_rel_+call_v
0.32::funky_a
0.32::bad_a
0.29::veteran_a
0.29::and_c+robot_n
0.28::and_c+criminal_n
0.28::bogus_a
0.28::talk_v+to_p+pron_rel_

0.27::investigate_v+murder_n
0.26::on_p+force_n
0.25::parody_n+of_p
0.25::Mason_n+and_c
0.25::pron_rel_+kill_v
0.25::racist_a
0.24::addicted_a
0.23::gritty_a
0.23::and_c+interference_n
0.23::arrive_v
0.23::and_c+detective_n
0.22::look_v+way_n
0.22::dead_a
0.22::pron_rel_+stab_v
0.21::pron_rel_+evade_v

The similarity of synonyms

- Similarity between *egglant/aubergine*: 0.11
  Relatively low cosine. Partly due to frequency (222 for *eggplant*, 56 for *aubergine*).
- Similarity between *policeman/cop*: 0.23
- Similarity between *city/town*: 0.73

In general, true synonymy does not correspond to higher similarity scores than near-synonymy.

Similarity of antonyms

- Similarities between:
  - cold/hot 0.29
  - dead/alive 0.24
  - large/small 0.68
  - colonel/general 0.33

Identifying antonyms

- Antonyms have high distributional similarity: hard to distinguish from near-synonyms purely by distributions.
- Identification by heuristics applied to pairs of highly similar distributions.
- For instance, antonyms are frequently coordinated while synonyms are not:
  - a selection of cold and hot drinks
  - wanted dead or alive

Distributions and knowledge

What kind of information do distributions encode?

- lexical knowledge
- world knowledge
- boundary between the two is blurry
- no perceptual knowledge

Distributions are partial lexical semantic representations, but useful and theoretically interesting.

Clustering

- clustering techniques group objects into clusters
- similar objects in the same cluster, dissimilar objects in different clusters
- allows us to obtain generalisations over the data
- widely used in various NLP tasks:
  - semantics (e.g. word clustering);
  - summarization (e.g. sentence clustering);
  - text mining (e.g. document clustering).

Distributional word clustering

We will:

- cluster words based on the contexts in which they occur
- assumption: words with similar meanings occur in similar contexts, i.e. are distributionally similar
- we will consider noun clustering as an example
- cluster 2000 nouns – most frequent in the British National Corpus
- into 200 clusters

Clustering nouns

truck lorry
bike
path
highway way
car
taxi
bicycle
street
road avenue
driver
house
mechanic
lab building
shack
engineer
scientist
office flat
plumber
writer
dwelling
journalist
proceedings
book
journal
newspaper
magazine

Clustering nouns

Feature vectors

- can use different kinds of context as features for clustering
  - window based context
  - parsed or unparsed
  - syntactic dependencies
- different types of context yield different results
- Example experiment: use verbs that take the noun as a direct object or a subject as features for clustering
- Feature vectors: verb lemmas, indexed by dependency type, e.g. subject or direct object
- Feature values: corpus frequencies

Extracting feature vectors: Examples

tree (Dobj)	**crop (Dobj)**	**tree (Subj)**	**crop (Subj)**
85 plant_v	76 grow_v	131 grow_v	78 grow_v
82 climb_v	44 produce_v	49 plant_v	23 yield_v
48 see_v	16 harvest_v	40 stand_v	10 sow_v
46 cut_v	12 plant_v	26 fell_v	9 fail_v
27 fall_v	10 ensure_v	25 look_v	8 plant_v
26 like_v	10 cut_v	23 make_v	7 spray_v
23 make_v	9 yield_v	22 surround_v	7 come_v
23 grow_v	9 protect_v	21 show_v	6 produce_v
22 use_v	9 destroy_v	20 seem_v	6 feed_v
22 round_v	7 spray_v	20 overhang_v	6 cut_v
20 get_v	7 lose_v	20 fall_v	5 sell_v
18 hit_v	6 sell_v	19 cut_v	5 make_v
18 fell_v	6 get_v	18 take_v	5 include_v
18 bark_v	5 support_v	18 go_v	5 harvest_v
17 want_v	5 see_v	18 become_v	4 follow_v
16 leave_v	5 raise_v	17 line_v	3 ripen_v
...	...	...	...

Feature vectors: Examples

tree
131 grow_v_Subj
85 plant_v_Dobj
82 climb_v_Dobj
49 plant_v_Subj
48 see_v_Dobj
46 cut_v_Dobj
40 stand_v_Subj
27 fall_v_Dobj
26 like_v_Dobj
26 fell_v_Subj
25 look_v_Subj
23 make_v_Subj
23 make_v_Dobj
23 grow_v_Dobj
22 use_v_Dobj
22 surround_v_Subj
22 round_v_Dobj
20 overhang_v_Subj
...

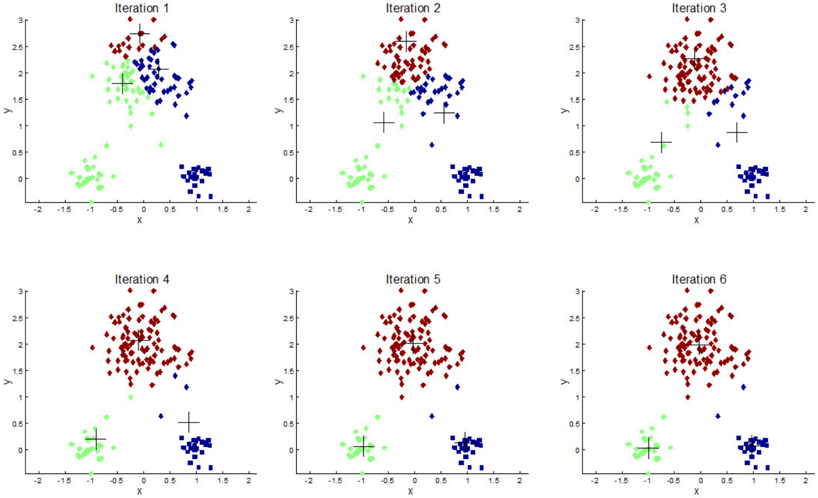
crop
78 grow_v_Subj
76 grow_v_Dobj
44 produce_v_Dobj
23 yield_v_Subj
16 harvest_v_Dobj
12 plant_v_Dobj
10 sow_v_Subj
10 ensure_v_Dobj
10 cut_v_Dobj
9 yield_v_Dobj
9 protect_v_Dobj
9 fail_v_Subj
9 destroy_v_Dobj
8 plant_v_Subj
7 spray_v_Subj
7 spray_v_Dobj
7 lose_v_Dobj
6 feed_v_Subj
...

Clustering algorithms, K-means

- many clustering algorithms are available
- example algorithm: K-means clustering
  - given a set of N data points $\{x_1, x_2, ..., x_N\}$
  - partition the data points into K clusters $C = \{C_1, C_2, ..., C_K\}$
  - minimize the sum of the squares of the distances of each data point to the cluster mean vector μ_i:

$$\arg\min_{C} \sum_{i=1}^{K} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{2}$$

K-means clustering



Noun clusters

tree crop flower plant root leaf seed rose wood grain stem forest garden
consent permission concession injunction licence approval
lifetime quarter period century succession stage generation decade phase interval future
subsidy compensation damages allowance payment pension grant
carriage bike vehicle train truck lorry coach taxi
official officer inspector journalist detective constable police policeman reporter
girl other woman child person people
length past mile metre distance inch yard
tide breeze flood wind rain storm weather wave current heat
sister daughter parent relative lover cousin friend wife mother husband brother father

Different senses of ***run***

The children **ran** to the store
If you see this man, **run**!
Service **runs** all the way to Cranbury
She is **running** a relief operation in Sudan
the story or argument **runs** as follows
Does this old car still **run** well?
Interest rates **run** from 5 to 10 percent
Who's **running** for treasurer this year?
They **ran** the tapes over and over again
These dresses **run** small

Subject arguments of ***run***

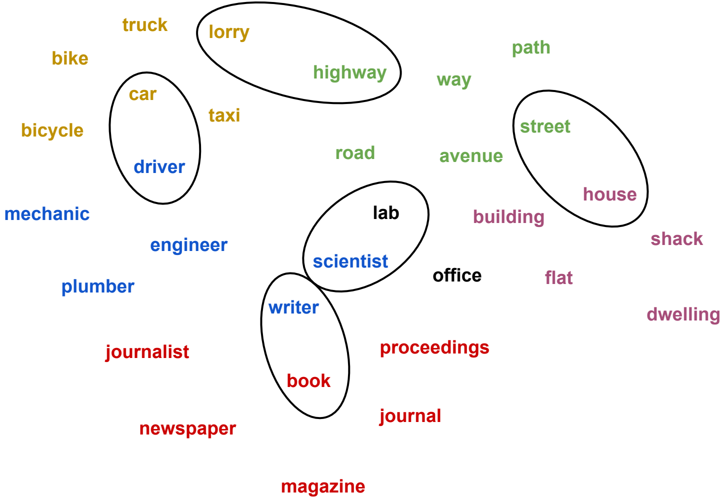
0.2125 drop tear sweat paint blood water juice
0.1665 technology architecture program system product version interface software tool computer network processor chip package
0.1657 tunnel road path trail lane route track street bridge
0.1166 carriage bike vehicle train truck lorry coach taxi
0.0919 tide breeze flood wind rain storm weather wave current heat
0.0865 tube lock tank circuit joint filter battery engine device disk furniture machine mine seal equipment machinery wheel motor slide disc instrument
0.0792 ocean canal stream bath river waters pond pool lake
0.0497 rope hook cable wire thread ring knot belt chain string
0.0469 arrangement policy measure reform proposal project programme scheme plan course
0.0352 week month year
0.0351 couple minute night morning hour time evening afternoon

Subject arguments of ***run*** (continued)

0.0341 criticism appeal charge application allegation claim objection suggestion case complaint
0.0253 championship open tournament league final round race match competition game contest
0.0218 desire hostility anxiety passion doubt fear curiosity enthusiasm impulse instinct emotion feeling suspicion
0.0183 expenditure cost risk expense emission budget spending
0.0136 competitor rival team club champion star winner squad county player liverpool partner leeds
0.0102 being species sheep animal creature horse baby human fish male lamb bird rabbit female insect cattle mouse monster
...

Clustering nouns

Clustering nouns



We can also cluster verbs...

sparkle glow widen flash flare gleam darken narrow flicker shine blaze bulge
gulp drain stir empty pour sip spill swallow drink pollute seep flow drip purify ooze pump bubble splash ripple simmer boil tread
polish clean scrape scrub soak
kick hurl push fling throw pull drag haul
rise fall shrink drop double fluctuate dwindle decline plunge decrease soar tumble surge spiral boom
initiate inhibit aid halt trace track speed obstruct impede accelerate slow stimulate hinder block
work escape fight head ride fly arrive travel come run go slip move

Uses of word clustering in NLP

Widely used in NLP as a source of lexical information:

- Word sense induction and disambiguation
- Modelling predicate-argument structure (e.g. semantic roles)
- Identifying figurative language and idioms
- Paraphrasing and paraphrase detection
- Used in applications directly, e.g. machine translation, information retrieval etc.

Distributional semantic models

1. Count-based models:
   - Explicit vectors: dimensions are elements in the context
   - **long sparse** vectors with **interpretable** dimensions
2. Prediction-based models:
   - Train a model to predict plausible contexts for a word
   - learn word representations in the process
   - **short dense** vectors with **latent** dimensions

Sparse vs. dense vectors

Why dense vectors?

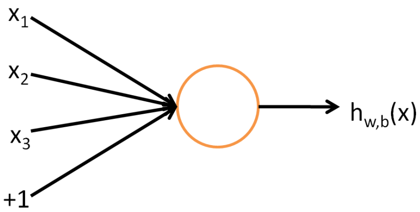
- easier to use as features in machine learning (less weights to tune)
- may generalize better than storing explicit counts
- may do better at capturing synonymy:
  - e.g. *car* and *automobile* are distinct dimensions in count-based models
  - will not capture similarity between a word with *car* as a neighbour and a word with *automobile* as a neighbour

Brief introduction to neural networks

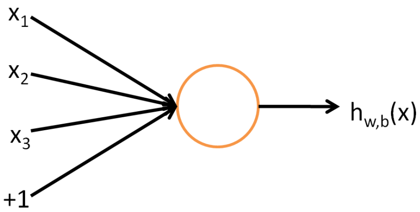
Supervised learning framework.

- Input: a set of labelled training examples $(x^{(i)}, y^{(i)})$
- Output: hypotheses $h_{W,b}(x)$ with parameters W, b which we fit to our data

The simplest possible neural network — single neuron



Neuron as a computational unit



$$h_{W,b}(x) = f(W^T x + b) = f(\sum_{i=1}^{3} W_i x_i + b)$$

where $f : \Re \mapsto \Re$ is the activation function,
W is a matrix of trainable weights,
b is the bias term.

Activation functions (common choices)

Sigmoid function

$$f(z) = \frac{1}{1 + e^{-z}}$$

output in range [0,1]

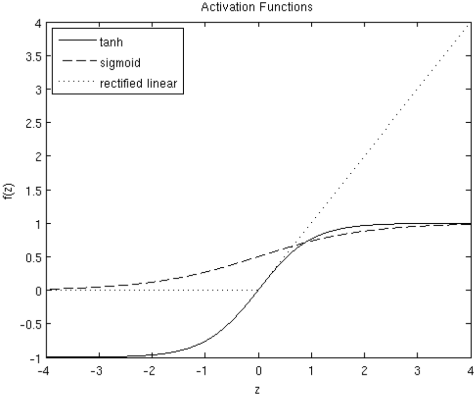
Hyperbolic tangent (tanh):

$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

output in range [-1,1]

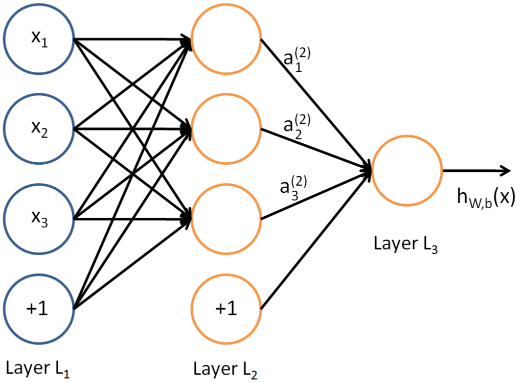
Rectified linear (ReLu):

$$f(z) = \max(0, z)$$



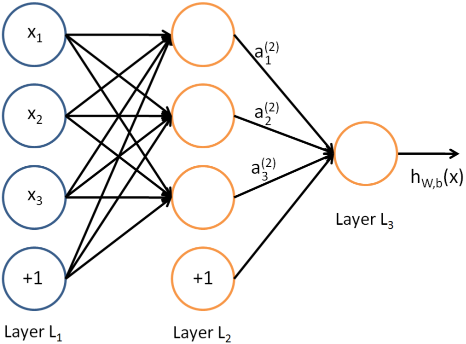
Multi-layer neural network

Feed-forward architecture



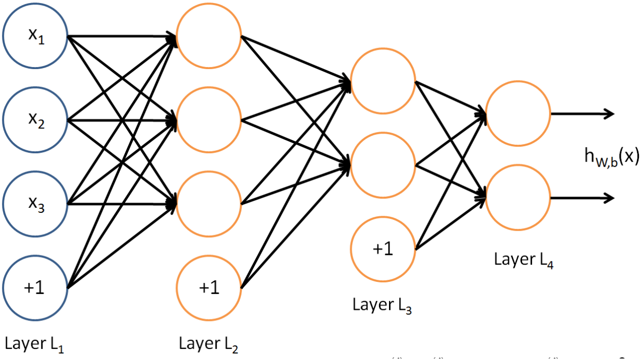
Input layer   Hidden layer   Output layer

Multi-layer neural network



$$z^{(2)} = W^{(1)}x + b^{(1)} \qquad a^{(2)} = f(z^{(2)})$$
$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \qquad h_{W,b}(x) = a^{(3)} = f(z^{(3)})$$

Deep neural networks and multi-class classification



Softmax function

Used in multi-class classification problems.

- Takes a vector of real values and squashes them into the range [0,1], so that they add up to 1
- use this as a **probability distribution** over output classes

$$softmax(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{d} e^{z_k}}$$

d is the dimensionality of the output layer

Acknowledgement

Some slides were adapted from Aurelie Herbelot

The introduction to neural networks is based on this helpful tutorial:
`http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/`