Welcome and Introduction

Philip Schulz and Wilker Aziz

https://github.com/philschulz/VITutorial

About us . . .

Wilker Aziz

- Research associate at UvA
- Sampling, VI, Machine Translation

Philip Schulz

- PhD candidate at UvA
- Applied Scientist at Amazon
- VI, Machine Translation, Bayesian Models

Problems

Supervised problems: "learn a distribution over observed data"

- sentences in natural language, images, videos, . . .

Unsupervised problems: "learn a distribution over observed and unobserved data"

- sentences in natural language + parse trees, images + bounding boxes . . .

Maximum likelihood estimation

We have data $x^{(1)}, \ldots, x^{(N)}$ e.g.

- sentences, images, ...

generated by some **unknown** procedure

Maximum likelihood estimation

We have data $x^{(1)}, \dots, x^{(N)}$ e.g.

- sentences, images, ...

generated by some **unknown** procedure
which we assume can be captured by a probabilistic model

- with **known** probability (mass/density) function e.g.

$$X \sim \mathrm{Cat}(\pi_1, \dots, \pi_K) \qquad \text{or} \qquad X \sim \mathcal{N}(\mu, \sigma^2)$$

Maximum likelihood estimation

We have data $x^{(1)}, \ldots, x^{(N)}$ e.g.

- sentences, images, ...

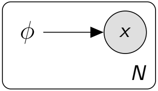
generated by some **unknown** procedure
which we assume can be captured by a probabilistic model

- with **known** probability (mass/density) function e.g.

$$X \sim \mathrm{Cat}(\pi_1, \ldots, \pi_K) \qquad \text{or} \qquad X \sim \mathcal{N}(\mu, \sigma^2)$$

and proceed to estimate parameters that assign maximum likelihood to observations

Multiple problems, same language



(Conditional) Density estimation

	ϕ	x
Parsing	a sentence	its syntactic/semantic parse tree/graph
Translation	a sentence	its translation
Captioning	an image	caption in English
Entailment	a text and hypothesis	entailment relation

Where does deep learning kick in?

Let ϕ be all side information available
e.g. deterministic *inputs/features*

Have neural networks predict parameters of our probabilistic model

$$X|\phi \sim \text{Cat}(\pi_w(\phi)) \quad \text{or} \quad X|\phi \sim \mathcal{N}(\mu_w(\phi), \sigma_w(\phi)^2)$$

and proceed to estimate parameters w of the NNs

Task-driven feature extraction

Often our side information ϕ is itself some high dimensional data

- ϕ is a sentence and x a tree
- ϕ is the source sentence and x is the target
- ϕ is an image and x is a caption

and part of the job of the NNs that parametrise our models is to also deterministically encode that input in a low-dimensional space

NN as efficient parametrisation

From the statistical point of view NNs do not generate data

- they parametrise distributions that *by assumption* govern data
- compact and efficient way to map from complex side information to parameter space

NN as efficient parametrisation

From the statistical point of view NNs do not generate data

- they parametrise distributions that *by assumption* govern data
- compact and efficient way to map from complex side information to parameter space

Prediction is done by a decision rule outside the statistical model

- e.g. beam search

MLE via gradient-based optimisation

The probability of an observation $X = x$ is given by some differentiable probability function

- the parameters of which are predicted by f_w *(also differentiable)*

MLE via gradient-based optimisation

The probability of an observation $X = x$ is given by some differentiable probability function

- the parameters of which are predicted by f_w *(also differentiable)*

Example: K classes

$$\mathrm{Cat}(X = x | \underbrace{f_1^K := f_w(\phi)}_{\text{class probabilities}}) = \prod_{i=1}^{K} f_i^{[x=i]}$$

MLE via gradient-based optimisation

The probability of an observation $X = x$ is given by some differentiable probability function

- the parameters of which are predicted by f_w *(also differentiable)*

Example: K classes

$$\mathrm{Cat}(X = x | \underbrace{f_1^K := f_w(\phi)}_{\text{class probabilities}}) = \prod_{i=1}^{K} f_i^{[x=i]}$$

Given a dataset of i.i.d. observations, SGD gives us a local optimum of the log-likelihood

DL in NLP recipe

Maximum likelihood estimation

- tells you which loss to optimise
  (i.e. negative log-likelihood)

Automatic differentiation (*backprop*)

- "give me a tractable forward pass and I will give you gradients"

Stochastic optimisation powered by backprop

- general purpose gradient-based optimisers

Tractability is central

Likelihood gives us a differentiable objective to optimise for

- but we need to stick with tractable likelihood functions

When do we have intractable likelihood?

Latent variables: assessing the likelihood requires marginalisation

- too many forward passes

$$P_X(x) = \sum_{c=1}^{K} \mathrm{Cat}(c|\pi_1, \ldots, \pi_K) \underbrace{\mathcal{N}(x|\mu_w(c), \sigma_w(c)^2)}_{\text{forward pass}}$$

When do we have intractable likelihood?

Latent variables: assessing the likelihood requires marginalisation

- too many forward passes

$$P_X(x) = \sum_{c=1}^{K} \mathrm{Cat}(c|\pi_1, \ldots, \pi_K) \underbrace{\mathcal{N}(x|\mu_w(c), \sigma_w(c)^2)}_{\text{forward pass}}$$

- even infinitely many

$$P_X(x) = \int \mathcal{N}(z|0, I) \underbrace{\mathrm{Cat}(x|\pi_w(z))}_{\text{forward pass}} \mathrm{d}z$$

Can we approximate the marginal?

Beam-search

- biased gradient estimates
  bye bye stochastic optimisation!

Can we approximate the marginal?

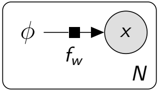
Beam-search

- biased gradient estimates
  bye bye stochastic optimisation!

Monte Carlo sampling

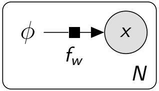
- breaks differentiability
  bye bye backprop!

What do we do then?



We know how to encode inductive bias through the design of the architecture

What do we do then?



We know how to encode inductive bias through the design of the architecture

But what if

- we want to learn clusters?
- or segmentation?
- or sparse models?
- or latent factors?
- or learn from incomplete supervision?
- or Bayesian NNs?

Deep Generative Models

Probabilistic models parametrised by neural networks

Deep Generative Models

Probabilistic models parametrised by neural networks

- better modelling assumptions
  one of the reasons why there's so much interest

Deep Generative Models

Probabilistic models parametrised by neural networks

- better modelling assumptions
  one of the reasons why there's so much interest
- but requires efficient inference

Deep Generative Models

Probabilistic models parametrised by neural networks

- better modelling assumptions
  one of the reasons why there's so much interest
- but requires efficient inference
  which is the reason why we are here today