Titanic

Datasettet

- Variabler fra 1309 passasjerer (eks. crew)
- pclass, survived, name, sex, age, sibsp, parch, ticket, fare, cabin, embarked, boat, body, home.dest
- http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic3info.txt
- Vi skal modellere survived!

Utfordringer

- Noen passasjerer mangler informasjon
- Variabler er alt fra float til strings
- Tallverdier utenfor [0,1]

Normalisering

- Læring kan bli ustabilt ved store verdier
- Normalisere: 0 i gjennomsnitt, 1 i standardavvik

```
    data = (data - data.mean()) / data.std()
    # see data.normalize
```

Manglende data

- Vi har foreslått å forkaste rader som mangler data
- Alle verdier blir likevel brukt til normalisering

```
    # in python, nan != nan.
    # the following line masks nan values
    data.mask = data != data
```

Diskret data

- pclass, parch, cabin, …
- Heltall mappes til one-hot vector: 2 -> [0,0,1,0]
- Strings mappes først til heltall, så til one-hot

```
    # convert raw value to integer
    c = mapper[raw]
    # set the corresponding one-hot value to 1
    dst[i][c] = 1
    # see data.to_one_hot
```

Oppgave 1

- Se i src/model.py#build
- Lag en modell! Ta gjerne utgangspunkt i mnist. (Si noe om skjulte lag.)
- Du har frie tøyler, men modellen må ha verdiene
  - y: output til nettverket
  - loss: tapet til nettverket (det vi prøver å minimere)
  - train: operasjon for å trene nettverket
  - input: eksempler fra datasett
  - ideal: ideelle verdier til datasett

Oppgave 2

- Se i src/data.py#titanic
- Velg interessante variabler du tror er nyttig
- Preprossesser disse ved behov (normalisering eller one-hot)

```
    fare = raw.normalize(‘fare')
    sexes = raw.to_one_hot(‘sex')
    x = np.ma.concatenate([fare, sexes], axis=1)
```

Oppgave 3

- Tweaking!
- Legg til ett eller flere skjulte lag
- Prøv andre variabler (cabin, anyone?)
- Bytt treningsalgoritme (optimizer)
- Dropout! Spør oss hva det er, sjekk API-et