Deep Learning

Exercise 10: Learn to Write like Shakespeare

Room: **BIN-1-B.01**
Instructor: Manuel Günther
Email: guenther@ifi.uzh.ch
Office: AND 2.54

Friday, May 14, 2021

Outline

Outline

Available Networks in PyTorch

Implementations for Recurrent Networks

- Full multi-layer recurrent networks
  - → `torch.nn.RNN` (Elman network), `torch.nn.LSTM`, `torch.nn.GRU`
  - → Parameters: `input_size`, `hidden_size`, `num_layers`, `bidirectional`
  - → Processes whole sequences, but less flexible
- Single recurrent cells (combines $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(r)}$ for RNN):
  - → `torch.nn.RNNCell`, `torch.nn.LSTMCell`, `torch.nn.GRUCell`
  - → Sequence processing needs to be done by hand

We will manually implement our own Elman network!

Text Data Processing

Data Processing

- Our data is text, i.e., sequence of characters
- Characters are transformed to one-hot encodings
  → `a` $\Rightarrow (1, 0, 0, \ldots, 0)$; `b` $\Rightarrow (0, 1, 0, \ldots, 0)$; $\ldots$
- Sequences of characters $\Rightarrow$ input matrix $\mathbf{X} \in \mathbb{R}^{S \times D}$
- Batch of sequences $\Rightarrow$ input tensor $\mathcal{X} \in \mathbb{R}^{B \times S \times D}$
  → Same sequence length S per batch, can vary between batches

Target Vectors

- One target for each character in the sequence
  → We want to predict the next character
- Zero-padding if sequence is longer than current index n

Text Data Processing

Example Encoding

- Original data: `abacac`
- Encodings: `a` $\Rightarrow (1,0,0)$; `b` $\Rightarrow (0,1,0)$; `c` $\Rightarrow (0,0,1)$;
- Get data for given index $n = 4$ for sequence length $S = 7$

  $\rightarrow$ Training sample `x = 00abac`, target sample `t = 0abaca`

  $\rightarrow$ Encodings: $\mathbf{X} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ $\mathbf{T} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$

Elman Network Implementation

Elman Network Details

- Weight matrices $\mathbf{W}^{(1)} \in \mathbb{R}^{K \times D}$, $\mathbf{W}^{(r)} \in \mathbb{R}^{K \times K}$, $\mathbf{W}^{(2)} \in \mathbb{R}^{D \times K}$
- Activation function g
- Input $\mathcal{X} \in \mathbb{R}^{B \times S \times D}$, output $\mathcal{Z} \in \mathbb{R}^{B \times S \times D}$

Elman Network Processing for Given $\mathcal{X}$

- Initialize hidden activation $\mathbf{H} \in \mathbb{R}^{B \times K} = 0$
- Iterate through sequence $s = 1, \ldots, S$:

$$\mathbf{A} \in \mathbb{R}^{B \times K} = \mathbf{W}^{(1)} \mathcal{X}_{:,s,:} + \mathbf{W}^{(r)} \mathbf{H}$$

$$\mathbf{H} \in \mathbb{R}^{B \times K} = g(\mathbf{A})$$

$$\mathcal{Z}_{:,s,:} = \mathbf{W}^{(2)} \mathbf{A}$$

Back Propagation Through Time

Categorical Cross-Entropy for BPTT

- Compare $\mathcal{Z} \in X \in \mathbb{R}^{B \times S \times D}$ with target $\mathcal{T} \in \mathbb{R}^{B \times S \times D}$
- SoftMax $\mathcal{Y} \in X \in \mathbb{R}^{B \times S \times D}$ from $\mathcal{Z} \in X \in \mathbb{R}^{B \times S \times D}$ over D
- Categorical cross-entropy loss over time:

$$\mathcal{J}^{\mathrm{CCE}} = -\frac{1}{B}\sum_{n=1}^{B}\frac{1}{S}\sum_{s=1}^{S}\sum_{o=1}^{O} t_o^{[n][s]} \log y_o^{[n][s]}$$

Particularities of `torch.nn.CrossEntropyLoss`

- SoftMax computed over **second** dimension
  - $\rightarrow$ Feature request submitted on PyTorch git
- We need SoftMax over **third** dimension (D) $\Rightarrow$ reorder matrices!

Outline

Shakespeare Poem

The Sonnets

- Poem by Shakespeare
- Around 2400 lines of text

Learn Text Characteristics

- Train Elman network to predict next character
- Apply to produce text from seed texts

Example Strophe

```
From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thy self thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.
```

Training Text URL

`http://raw.githubusercontent.com/brunoklein99/deep-learning-notes/master/shakespeare.txt`

Data and Targets

Task 1: Data Characteristics

- Load the data from the text file
  - → Decide how to handle newline `\n` characters
- Get the number of unique characters in the text (D)

Task 2: One-hot Encoding

- For each unique character, provide one-hot encoding vector
- Store in a dictionary: character ⇒ encoding vector

Data and Targets

Task 3: Sequence Coding

- Implement a function that provides $\mathbf{X}^{[n]}, \mathbf{T}^{[n]}$ both $\in \mathbb{R}^{S \times D}$
  - $\rightarrow$ Index n for our original data, variable sequence length S
  - $\rightarrow$ Apply zero-padding as described

Calculation of Indexes for n and S

$$\mathbf{X}^{[n]} = \{\text{enc}(n - S + s - 1) | 1 \leq s \leq S\} \qquad \mathbf{T}^{[n]} = \{\text{enc}(n - S + s) | 1 \leq s \leq S\}$$

Test 1: Sequences

- Get inputs and targets for $n = 2$ and $S = 5$
- Check that the zero-padding in the beginning is correct
- Check that last entries are one-hot vectors

Data and Targets

Task 4: Dataset and Data Loader

- Implement `torch.utils.data.Dataset` for our data
- Constructor `__init__(self, data, S)` takes data and S
  - $\rightarrow$ Store current sequence length S as variable `self.S`
- Index function `__getitem__(self, index)` returns $(\mathbf{X}^{[n]}, \mathbf{T}^{[n]})$ for n
- Length `__len__(self)` returns the number of sequences
- Instantiate `torch.utils.data.DataLoader` from dataset

Test 2: Data Sizes

- Iterate though all batches in the dataset
- Check that batches $\mathcal{X}$ and $\mathcal{T}$ are in correct size
- Check that content of $\mathcal{X}$ and $\mathcal{T}$ is related

Elman Network Training

Task 5: Elman Network Implementation

- Implement Elman network:
  - → First layer $\mathbf{W}^{(1)}$, recurrent layer $\mathbf{W}^{(r)}$ and second layer $\mathbf{W}^{(2)}$, PReLU
- Instantiate fully-connected layers for given K and D
- Implement `forward` function sequentially on S, parallel on B
- Return logits $\mathcal{Z}$ for input $\mathcal{X}$

Test 3: Network Output

- Instantiate Elman network with arbitrary D and K
- Generate artificial batch of samples $\mathcal{X}$
- Check that output $\mathcal{T}$ is in desired shape

Elman Network Training

Task 6: Training Loop

- Instantiate Elman network with D from data and $K = 1000$
- Use Adam optimizer with learning rate η
- Use categorical cross-entropy loss
  - $\rightarrow$ Make sure that SoftMax is executed over correct dimension
- Compute training loss over epoch
- Change `dataset.S` randomly after each batch to $S \in [5, 20]$
- Train for 10 epochs (or more)

Writing a Poem

Task 7: Text Encoding

- Implement function `encode(text)`
  - → Turn `text` of length S into $\mathcal{X} \in \mathbb{R}^{1 \times S \times D}$

Task 8: Next Element Prediction

- Implement function `predict(logits)`
  - → Take logit of **last** sequence item $\vec{y}^{\{S\}}$
  - → Option 1: Return character with the highest confidence
  - → Option 2: Sample character based on confidences

Writing a Poem

Task 9: Sequence Completion

- Take seed text to start from
- Iteratively add characters to current text
- Stop after 80 characters have been added (can be more)

Task 10: Text Production

- Define several seeds (e.g., `"th"`, `"moth"`, `"lov"`)
- Complete sequence with option 1 (highest probability)
- Complete sequence with option 2 (sampled probability)
- Write results to console

Writing a Poem

Exemplary Outputs for Some Seeds

seed	best output	sampled output
th	the story of thee this strangely pass, and scarcely greatst converted from the sta	thou is his diss and and meant the deach of beauty os my loves as end meas. form a
beau	beauty thou wilt take, thou best of deains the tillage of the truth'n of both, and t	beauty shows the worst was thy humour doth and with heat, no, i that power ald are,
mothe	mother's glass hid err that thou shalt find thus makes but cours morr worts and to hi	mother's would by ill believing thingsteration of hers buttone, cansed of self grow.
bloo	blood which youngly those that stell of good, or evil luck, of plagues, of deaves sh	blooms have full as deeptate of youth in each st that i come so near, sweet to thee
q	quite, for i me deathroom thou wilt swift-foot shale rone, nor east sure i heat a	quity good a poor heam and men. yet this might, and by and by clespest of my pime
wh	when i against my self with thee shall not be tomb ex then seems? thou the beauty	when i (perhaps) compound so thou bear'st love to any whe hash on love to thy and