Deep Sequential Models

NewLEADS
31 Oct 2019

Carl Andersson

Uppsala University, Department of Information Technology
Introduction

▶ Cancer classification from images
Introduction

- Cancer classification from images
- Machine translation
Introduction

▶ Cancer classification from images
▶ Machine translation
▶ Natural speech generation
Introduction

- Cancer classification from images
- Machine translation
- Natural speech generation
- Music generation
Introduction

- Cancer classification from images
- Machine translation
- Natural speech generation
- Music generation
- Superhuman players in Go, Chess and Starcraft 2
Introduction

- Cancer classification from images
- Machine translation
- Natural speech generation
- Music generation
- Superhuman players in Go, Chess and Starcraft 2
- <Insert arbitrary data here>
Introduction

- Cancer classification from images
- Machine translation
- Natural speech generation
- Music generation
- Superhuman players in Go, Chess and Starcraft 2
- <Insert arbitrary data here>

Why can it be used so generally?
What is the meaning of Deep?

- Hierarchical feature extraction
What is the meaning of Deep?

- Hierarchical feature extraction
- Low level, conceptually easy, features ➞ High level features
What is the meaning of Deep?

- Hierarchical feature extraction
- Low level, conceptually easy, features $\rightarrow$ High level features
- Does not have to be neural networks!\(^1\)

---

What is the meaning of Deep?

- Hierarchical feature extraction
- Low level, conceptually easy, features $\Rightarrow$ High level features
- Does not have to be neural networks!\(^1\)
- A black box function approximator

What is the meaning of Deep?

- Hierarchical feature extraction
- Low level, conceptually easy, features $\Rightarrow$ High level features
- Does not have to be neural networks!\(^1\)
- A black box function approximator

But is the box really that black?

Problem formulation

What?
- Sequential data

```latex
t_{-1} \rightarrow t \rightarrow t+1 \rightarrow t+2 \rightarrow ...
```

Why?
- Control/reinforcement learning
Problem formulation

What?
▶ Sequential data
▶ Long and short correlations

Why?
▶ Control/reinforcement learning

\[ z_{t-1}, z_t, z_{t+1}, z_{t+2}, \ldots \]
\[ x_{t-1}, x_t, x_{t+1}, x_{t+2}, \ldots \]
\[ u_{t-1}, u_t, u_{t+1}, u_{t+2}, \ldots \]
Problem formulation

What?
▶ Sequential data
▶ Long and short correlations

How?
▶ Deep learning
▶ One-step-ahead predictor

Why?
▶ Control/reinforcement learning

\[ z_{t-1} \rightarrow z_t \rightarrow z_{t+1} \rightarrow z_{t+2} \rightarrow \ldots \]
\[ x_{t-1} \rightarrow x_t \rightarrow x_{t+1} \rightarrow x_{t+2} \rightarrow \ldots \]
\[ u_{t-1} \rightarrow u_t \rightarrow u_{t+1} \rightarrow u_{t+2} \rightarrow \ldots \]
Problem formulation

What?
▶ Sequential data
▶ Long and short correlations

How?
▶ Deep learning

\[ z_{t-1} \rightarrow z_t \rightarrow z_{t+1} \rightarrow z_{t+2} \rightarrow \ldots \]
\[ u_{t-1} \rightarrow u_t \rightarrow u_{t+1} \rightarrow u_{t+2} \rightarrow \ldots \]
\[ x_{t-1} \rightarrow x_t \rightarrow x_{t+1} \rightarrow x_{t+2} \rightarrow \ldots \]
Problem formulation

What?
- Sequential data
- Long and short correlations

How?
- Deep learning
- One-step-ahead predictor
Problem formulation

What?
- Sequential data
- Long and short correlations

How?
- Deep learning
- One-step-ahead predictor

Why?

\[ z_t \rightarrow z_{t+1} \rightarrow z_{t+2} \rightarrow \ldots \]

\[ x_t \rightarrow x_{t+1} \rightarrow x_{t+2} \rightarrow \ldots \]

\[ u_t \rightarrow u_{t+1} \rightarrow u_{t+2} \rightarrow \ldots \]
Problem formulation

What?
▶ Sequential data
▶ Long and short correlations

How?
▶ Deep learning
▶ One-step-ahead predictor

Why?
▶ Control/reinforcement learning
Outline

- What is the meaning of Deep?
- Problem formulation
- Deep Learning: Images
- Deep learning: Sequences
- Examples
- Conclusion
- Future
Deep learning: Images

- Correlations are (mainly) local
- Pixels $\rightarrow$ Edges $\rightarrow$ Objects
Deep learning: Images

- Correlations are (mainly) local
- Pixels -> Edges -> Objects

Convolutional Neural Network
Deep learning: Images

- Alternating local convolutions with down sampling

Deep learning: Images

- Alternating local convolutions with down sampling
- The model is a strong prior
Deep learning: Images

- Correlations are (mainly) local
- Pixels -> Edges -> Objects

Convolutional Neural Network

- Prior for local correlations
- Prior also for non-local correlations
Deep learning: Sequences

- Correlations are (mainly) local
- Causality
- Character -> Word -> Sentence
- Moving a leg -> Taking steps -> Dancing
Deep learning: Sequences

- Correlations are (mainly) local
- Causality
- Character -> Word -> Sentence
- Moving a leg -> Taking steps -> Dancing

Recurrent Neural Network
Long short-term memory model

\[ p(x_{1:T} | u_{1:T}) = \prod_{t} p(x_t | x_{1:t-1}, u_{1:t}) \approx \prod_{t} p(x_t | s_t) \]

▶ A nonlinear state space model

---

Long short-term memory model$^2$

$$p(x_{1:T} \mid u_{1:T}) = \prod_t p(x_t \mid x_{1:t-1}, u_{1:t}) \approx \prod_t p(x_t \mid s_t)$$

- A nonlinear state space model
- The state is a statistic that is recursively updated

$$s_t = g(s_{t-1}, u_t) \quad \text{or} \quad s_t = g(s_{t-1}, [x_{t-1}, u_t])$$

---

Long short-term memory model

\[ p(x_{1:T} | u_{1:T}) = \prod_t p(x_t | x_{1:t-1}, u_{1:t}) \approx \prod_t p(x_t | s_t) \]

- A nonlinear state space model
- The state is a statistic that is recursively updated

\[ s_t = g(s_{t-1}, u_t) \quad \text{or} \quad s_t = g(s_{t-1}, [x_{t-1}, u_t]) \]

- \( g \) is designed to be trainable with backpropagation and for long data dependencies

---

Long short-term memory model\(^2\)

\[
p(x_{1:T} | u_{1:T}) = \prod_t p(x_t | x_{1:t-1}, u_{1:t}) \approx \prod_t p(x_t | s_t)
\]

- A nonlinear state space model
- The state is a statistic that is recursively updated

\[
s_t = g(s_{t-1}, u_t) \quad \text{or} \quad s_t = g(s_{t-1}, [x_{t-1}, u_t])
\]

- \(g\) is designed to be trainable with backpropagation and for long data dependencies
- Missing depth(?)

---

Long short-term memory model

\[
\begin{align*}
&u_{t-1} & u_t & u_{t+1} & u_{t+2} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
&s_{t-1} & s_t & s_{t+1} & s_{t+2} \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
x_{t-1} & x_t & x_{t+1} & x_{t+2} \\
\end{align*}
\]
Long short-term memory model

\[ \begin{align*}
&\ldots \\
&u_{t-1} \\
&\downarrow \\
&s_{t-1} \\
&\downarrow \\
&x_{t-1} \\
&\ldots \\
&u_t \\
&\downarrow \\
&s_t \\
&\downarrow \\
&x_t \\
&\ldots \\
&u_{t+1} \\
&\downarrow \\
&s_{t+1} \\
&\downarrow \\
&x_{t+1} \\
&\ldots \\
&u_{t+2} \\
&\downarrow \\
&s_{t+2} \\
&\downarrow \\
&x_{t+2} \\
&\ldots 
\end{align*} \]
Example: Text modeling

Model text with LSTM

- The LSTM predicts the next character in a sequence
- No exogenous input
Example: Text modeling

Model text with LSTM

▶ The LSTM predicts the next character in a sequence
▶ No exogenous input

Experiments from Karpathy³

▶ Shakespeare
▶ Latex papers
▶ Linux kernel source

³http://karpathy.github.io/2015/05/21/rnn-effectiveness
Example: Text modeling

The model learns:

▶ to spell words

KING LEAR: O, if you were a feeble sight, the courtesy of your law, Your sight and several breath, will wear the gods With his heads, and my hands are wonder’d at the deeds, So drop upon your lordship’s head, and your opinion Shall be against your honour.
Example: Text modeling

The model learns:

▶ to spell words
▶ latex environments

\begin{proof}
Omitted.
\end{proof}

\textbf{Lemma 0.1.} Let \( \mathcal{C} \) be a set of the construction.
Let \( \mathcal{C} \) be a gerber covering. Let \( \mathcal{F} \) be a quasi-coherent sheaves of \( \mathcal{O} \)-modules. We have to show that
\[
\mathcal{O}_{\mathcal{C}} = \mathcal{O}_{\mathcal{C}}(\mathcal{L})
\]
\[.
\]
\textbf{Proof.} This is an algebraic space with the composition of sheaves \( \mathcal{F} \) on \( \mathcal{X}_{\text{étale}} \) we have
\[
\mathcal{O}_{\mathcal{X}}(\mathcal{F}) = \{ \text{morphisms } \times \mathcal{O}_{\mathcal{X}}(\mathcal{G}, \mathcal{F}) \}
\]
where \( \mathcal{G} \) defines an isomorphism \( \mathcal{F} \to \mathcal{F} \) of \( \mathcal{O} \)-modules. \( \square \)

\textbf{Lemma 0.2.} This is an integer \( Z \) is injective.
\textbf{Proof.} See Spaces, Lemma ??.
\( \square \)

\textbf{Lemma 0.3.} Let \( S \) be a scheme. Let \( X \) be a scheme and \( X \) is an affine open covering. Let \( U \subseteq X \) be a canonical and locally of finite type. Let \( X \) be a scheme. Let \( X \) be a scheme which is equal to the formal complex.
The following to the construction of the lemma follows.
\textbf{Proof.} Let \( X \) be a scheme covering. Let
\[
b : X \to Y' \to Y \to Y' \times_X Y \to X.
\]
be a morphism of algebraic spaces over \( S \) and \( Y \).
\textbf{Proof.} Let \( X \) be a nonzero scheme of \( X \). Let \( X \) be an algebraic space. Let \( \mathcal{F} \) be a quasi-coherent sheaf of \( \mathcal{O}_X \)-modules. The following are equivalent
\begin{enumerate}
\item \( \mathcal{F} \) is an algebraic space over \( S \).
\item If \( X \) is an affine open covering.
\end{enumerate}
Consider a common structure on \( X \) and \( X \) the functor \( \mathcal{O}_X(U) \) which is locally of finite type. \( \square \)
Example: Text modeling

The model learns:

- to spell words
- \LaTeX{} environments
- to keep track of scopes

```c
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (!state)
        cmd = (int)(int_state^((in_8(&ch->ch_flags) & Cmd) ? 2 : 1));
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << 1))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                    ((count & 0x00000000ffffff) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    } /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy");
}
```
Example: Text modeling

The model learns:

- to spell words
- \LaTeX\ environments
- to keep track of scopes

Hard to train and optimize

static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTHREAD_UNCCA) +
            ((count & 0x00000000fffffff8) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_contols(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy");
}
Deep learning: Sequences

- Correlations are (mainly) local
- Causality
- Character -> Word -> Sentence
- Moving a leg -> Taking steps -> Dancing

Recurrent Neural Network

- Causal
- Prior for short term correlations
- Does not impose a deep prior(?)
Deep learning: Sequences

- Correlations are (mainly) local
- Causality
- Character -> Word -> Sentence
- Moving a leg -> Taking steps -> Dancing

Recurrent Neural Network
- Causal
- Prior for short term correlations
- Does not impose a deep prior (?)

Temporal convolutional network
Temporal convolutional network

$p(x_{1:T} | u_{1:T}) = \prod_{t} p(x_t | x_{1:t-1}, u_{1:t}) \approx \prod_{t} p(x_t | x_{t-K-1:t-1}, u_{t-K:t})$

▶ A non-linear ARX model

---

Temporal convolutional network

\[
p(x_{1:T} \mid u_{1:T}) = \prod_t p(x_t \mid x_{1:t-1}, u_{1:t}) \approx \prod_t p(x_t \mid x_{t-K-1:t-1}, u_{t-K:t})
\]

- A non-linear ARX model
- Parameterize \( p(x_t \mid x_{t-K-1:t-1}, u_{t-K:t}) \) with a deep convolutional network

---

Temporal convolutional network

\[ p(x_{1:T} \mid u_{1:T}) = \prod_{t} p(x_t \mid x_{1:t-1}, u_{1:t}) \approx \prod_{t} p(x_t \mid x_{t-K-1:t-1}, u_{t-K:t}) \]

- A non-linear ARX model
- Parameterize \( p(x_t \mid x_{t-K-1:t-1}, u_{t-K:t}) \) with a deep convolutional network
- Can efficiently be parallelized

---

Temporal convolutional network
Temporal convolutional network

\[ x_{t-3} \quad x_{t-2} \quad x_{t-1} \quad x_t \nabla_t \quad u_{t-2} \quad u_{t-1} \quad u_t \nabla_t \quad u_{t+1} \]

\[ h_{t-1} \quad h_{t+1} \]

\[ h_{t+1}^2 \]
Temporal convolutional network
Temporal convolutional network

\[ \hat{x}_{t-4} \hat{x}_{t-3} \hat{x}_{t-2} \hat{x}_{t-1} \hat{x}_t \hat{x}_{t+1} \]

\[ h^1_{t-4} h^1_{t-3} h^1_{t-2} h^1_{t-1} h^1_t h^1_{t+1} \]

\[ h^2_{t-4} h^2_{t-3} h^2_{t-2} h^2_{t-1} h^2_t h^2_{t+1} \]

\[ \cdots \]
Potential problem?

Short term correlations for one step ahead predictions

Autocorrelation for the text on KTH about page
Example: Silverbox

- An electrical circuit
- Non-linear
- No long term correlations
- Little to no process noise

\[ u \rightarrow x \]

---

Example: Silverbox

Prediction and error for a TCN model

Example: Silverbox

<table>
<thead>
<tr>
<th>RMSE (mV)</th>
<th>Which samples</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>first 25 000</td>
<td>Local Linear State Space</td>
</tr>
<tr>
<td>0.24</td>
<td>first 30 000</td>
<td>NLSS with sigmoids</td>
</tr>
<tr>
<td>1.9</td>
<td>400 to 30 000</td>
<td>Wiener-Schetzen</td>
</tr>
<tr>
<td>0.31</td>
<td>first 25 000</td>
<td>LSTM</td>
</tr>
<tr>
<td>0.58</td>
<td>first 30 000</td>
<td>LSTM</td>
</tr>
<tr>
<td>0.75</td>
<td>first 25 000</td>
<td>MLP</td>
</tr>
<tr>
<td>0.95</td>
<td>first 30 000</td>
<td>MLP</td>
</tr>
<tr>
<td>0.75</td>
<td>first 25 000</td>
<td>TCN</td>
</tr>
<tr>
<td>1.16</td>
<td>first 30 000</td>
<td>TCN</td>
</tr>
</tbody>
</table>

RMSE for free run simulation

- LSTM better than TCN
Example: F-16 Ground Vibration Test

- Forces on ordnances on an F-16 from shaking
- Non-linear due to structural dynamics
- No long term correlations
- Small amount of process noise

Example: F16

<table>
<thead>
<tr>
<th>Mode</th>
<th>LSTM</th>
<th>MLP</th>
<th>TCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free-run simulation</td>
<td>0.74</td>
<td>0.48</td>
<td>0.63</td>
</tr>
<tr>
<td>One-step-ahead prediction</td>
<td>0.023</td>
<td>0.045</td>
<td>0.034</td>
</tr>
</tbody>
</table>

RMSE for F16 data

- Improvement versus linear models and previous attempted nonlinear models
- LSTM and TCN equally good

---

Deep learning: Sequences

- Correlations are (mainly) local
- Character \(\rightarrow\) Word \(\rightarrow\) Sentence
- Causality

Recurrent Neural Network

- Causal
- Prior for short term correlations
- Does not impose a deep prior(?)

Temporal convolutional network

- Causal
- A deep prior
- Does not impose a short term prior for predictions
LSTM vs TCN

▶ TCN are easier to optimize
▶ LSTM can find a better model

Conclusion

- LSTM & TCN both have advantages
- TCN has a stronger deep prior
- LSTM has a stronger short term prediction prior
Conclusion

▶ LSTM & TCN both have advantages
▶ TCN has a stronger deep prior
▶ LSTM has a stronger short term prediction prior

Possible to combine?
On going work

What if a single mode is not enough?

---

On going work

▶ What if a single mode is not enough?
▶ Combine the temporal models with variational autoencoder

---

On going work

▶ What if a single mode is not enough?
▶ Combine the temporal models with variational autoencoder
▶ E.g. Stochastic TCN\textsuperscript{10}

Stochastic TCN
Preliminary results

- Trained on classical piano music in midi format

Hopefully we can do even better
Preliminary results

- Trained on classical piano music in midi format
- Evaluated with log likelihood on one step ahead prediction
Preliminary results

- Trained on classical piano music in midi format
- Evaluated with log likelihood on one step ahead prediction
- Reaches state of the art for this dataset
Preliminary results

- Trained on classical piano music in midi format
- Evaluated with log likelihood on one step ahead prediction
- Reaches state of the art for this dataset

Hopefully we can do even better
Thanks for listening

Questions?