Casting deep nets on financial crime

Sequence Models for Fraud

September 17th, 2019

João Tiago Ascensão
joao.ascensao@feedzai.com
t.me/jtascensao

feedzai research

About me

João Tiago Ascensão

Director of Data Science at Feedzai Research
Founding member, Head of Curriculum at the LDSA

joao.ascensao@feedzai.com
t.me/jtascensao
@jtascensao

Agenda

1. Motivation
2. Problem Statement
3. Option A - Sequence Features
4. Option B - Sequence Models
5. Problem Solution
6. Takeaways

Appendix - Practical Resources

Motivation

Motivation

- Take the following **tabular data**:

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	1:30pm	Fraud
4	C	Website2	200.00	1:31pm	Fraud
5	C	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- Imagine we want to classify the dummy transactions as fraud or legit.

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	1:30pm	Fraud
4	C	Website2	200.00	1:31pm	Fraud
5	C	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- The three transactions at the bottom are fraud, are they suspicious somehow?

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	1:30pm	Fraud
4	C	Website2	200.00	1:31pm	Fraud
5	C	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- The **same card**...

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	**C**	Website1	200.00	1:30pm	Fraud
4	**C**	Website2	200.00	1:31pm	Fraud
5	**C**	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- The same card, in **three merchants**...

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	**Website1**	200.00	1:30pm	Fraud
4	C	**Website2**	200.00	1:31pm	Fraud
5	C	**Website3**	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- The same card, in three merchants, for **same amount**...

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	**200.00**	1:30pm	Fraud
4	C	Website2	**200.00**	1:31pm	Fraud
5	C	Website3	**200.00**	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- The same card, in three merchants, for same amount, in **three minutes**.

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	**1:30pm**	Fraud
4	C	Website2	200.00	**1:31pm**	Fraud
5	C	Website3	200.00	**1:32pm**	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- When we feed this into a classifier, is it able to pick up such patterns?

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	1:30pm	Fraud
4	C	Website2	200.00	1:31pm	Fraud
5	C	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- No, because each transaction is looked at **independently**, regardless of history.

	Card	Merchant	Amount USD	Hour	Label
1	A	Pizza Place	10.00	1:15pm	Legitimate
2	B	Coffee Place	5.00	1:18pm	Legitimate
3	C	Website1	200.00	1:30pm	Fraud
4	C	Website2	200.00	1:31pm	Fraud
5	C	Website3	200.00	1:32pm	Fraud

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- But yes, using **feature engineering** and **profiles** that convey past information.

	Card	Merchant	Amount USD	Hour	Label	Trx by card
1	A	Pizza Place	10.00	1:15pm	Legitimate	1
2	B	Coffee Place	5.00	1:18pm	Legitimate	1
3	C	Website1	200.00	1:30pm	Fraud	1
4	C	Website2	200.00	1:31pm	Fraud	2
5	C	Website3	200.00	1:32pm	Fraud	3

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- Profiles involve **aggregations by entity over sliding windows of X time units**.

	Card	Merchant	Amount USD	Hour	Label	Trx by card
1	A	Pizza Place	10.00	1:15pm	Legitimate	1
2	B	Coffee Place	5.00	1:18pm	Legitimate	1
3	C	Website1	200.00	1:30pm	Fraud	1
4	C	Website2	200.00	1:31pm	Fraud	2
5	C	Website3	200.00	1:32pm	Fraud	3

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Motivation

- Feature engineering is non-trivial, expensive and requires domain knowledge
- **Feedzai's AutoML** computes hundred of profiles before training a model
- But it takes time and requires managing profiles and related state in production
- Feature engineering is good, no feature engineering is better.

Problem Statement

Problem statement

- Consider a supervised learning setting, with labeled training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$

Problem statement

- Consider a supervised learning setting, with labeled training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$
- Each training instance is comprised of features $\boldsymbol{x}_i = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\}$
- Features can be numerical, categorical, timestamps, or identification fields

Problem statement

- Consider a supervised learning setting, with labeled training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$
- Each training instance is comprised of features $\boldsymbol{x}_i = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\}$
- Features can be numerical, categorical, timestamps, or identification fields
- And a training binary label $y_i \in \{0, 1\}$

Problem statement

- Consider a supervised learning setting, with labeled training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$
- Each training instance is comprised of features $\boldsymbol{x}_i = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\}$
- Features can be numerical, categorical, timestamps, or identification fields
- And a training binary label $y_i \in \{0, 1\}$
- We train a model to estimate the probability of fraud $\hat{y}_i = h(x_i) = P(y_i = 1 | x_i)$

Problem statement

- Consider a supervised learning setting, with labeled training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$
- Each training instance is comprised of features $\boldsymbol{x}_i = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}\}$
- Features can be numerical, categorical, timestamps, or identification fields
- And a training binary label $y_i \in \{0, 1\}$
- We train a model to estimate the probability of fraud $\hat{y}_i = h(x_i) = P(y_i = 1 | x_i)$
- And make decisions (approve, block) given a probability threshold, y_{thr}.

feedzai

Reframing for sequences

- The assumption is that the history of an entity conveys important information
- This means that the probability of fraud depends on the current and past events

Reframing for sequences

- The assumption is that the history of an entity conveys important information
- This means that the probability of fraud depends on the current and past events
- Hence, we adapt this probability as:

$$P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{1,k})$$

- Corresponding to the the i-th transaction from the k-th card

Reframing for sequences

- The assumption is that the history of an entity conveys important information
- This means that the probability of fraud depends on the current and past events
- Hence, we adapt this probability as:

$$P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{1,k})$$

- Corresponding to the the i-th transaction from the k-th card
- The history is described by all past events of the entity

Reframing for sequences

- The assumption is that the history of an entity conveys important information
- This means that the probability of fraud depends on the current and past events
- Hence, we adapt this probability as:

$$P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \dots, x_{1,k})$$

- Corresponding to the the i-th transaction from the k-th card
- The history is described by all past events of the entity
- The time difference between steps, δt, is not constant.

Truncating sequences

- Sometimes it might not be feasible to consider the entire history
- It might also not be entirely desirable and too strong an assumption
- We adapt the probability to be more general, relative to a number of steps, T:

$$P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k})$$

- In short, we look back T steps back in time.

Option A

Sequence Features

Defining feature engineering

- Feature engineering consists in determining which features might be useful

Defining feature engineering

- Feature engineering consists in determining which features might be useful
- It comprises selecting, adapting, and creating new features, iteratively
- And converting them to a convenient format, typically real-numbered

$$z_i = f(x_i)$$

Defining feature engineering

- Feature engineering consists in determining which features might be useful
- It comprises selecting, adapting, and creating new features, iteratively
- And converting them to a convenient format, typically real-numbered

$$z_i = f(x_i)$$

- This new representation will then be fed to the model, connected in a chain

$$\hat{y}_i = h(z_i) = h(f(x_i))$$

Feature engineering sequences

- Given our assumption that $P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k})$
- We need to inject past information into our classifier

Feature engineering sequences

- Given our assumption that $P(y_{i,k} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k})$
- We need to inject past information into our classifier
- Therefore, we generate features based on current and the T past transactions

$$z_{i,k} = f(x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k})$$

Classification task

- This representation, enriched feature vectors, is used as input for the classifier

$$\hat{y}_{i,k} = h(z_{i,k}) = h(f(x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k}))$$

- It can be any non-sequence based classification method (e.g., gradient boosting)

Classification task

- This representation, enriched feature vectors, is used as input for the classifier

$$\hat{y}_{i,k} = h(z_{i,k}) = h(f(x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k}))$$

- It can be any non-sequence based classification method (e.g., gradient boosting)
- Time dependency is fully captured by the features

Classification task

- This representation, enriched feature vectors, is used as input for the classifier

$$\hat{y}_{i,k} = h(z_{i,k}) = h(f(x_{i,k}, x_{i-1,k}, \ldots, x_{i-T,k}))$$

- It can be any non-sequence based classification method (e.g., gradient boosting)
- Time dependency is fully captured by the features
- This requires the availability (or else, computing) of profiles before scoring.

Option B

Sequence Models

Deep feedforward networks

- Information flows one-way, through the intermediate computations to the output

Deep feedforward networks

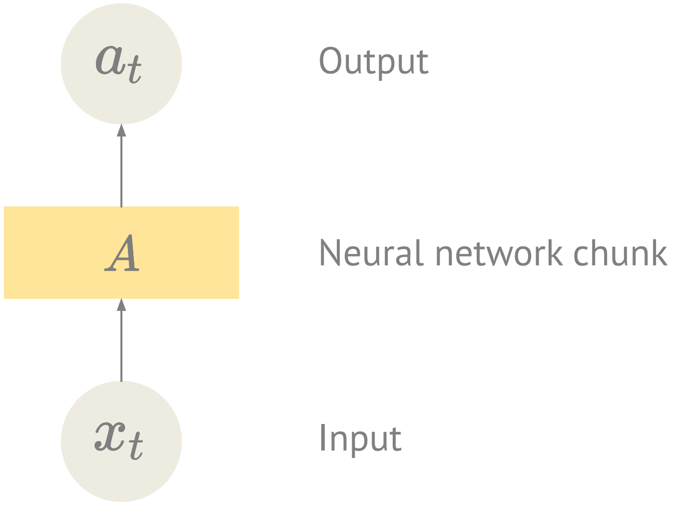
- Information flows one-way, through the intermediate computations to the output
- No feedback connections in which information of the model is fed back into it

Deep feedforward networks

- Information flows one-way, through the intermediate computations to the output
- No feedback connections in which information of the model is fed back into it
- Stateless, there is no built-in memory about past events.

Deep feedforward networks

- Information flows one-way, through the intermediate computations to the output
- No feedback connections in which information of the model is fed back into it
- Stateless, there is no built-in memory about past events.



Introducing statefulness

- Again, we assume that the probability of fraud depends on past events

$$P(y_{k,i} = 1 | x_{i,k}, x_{i-1,k}, \ldots, x_{1,k})$$

- But consider the case in which we condense the past knowledge in state s

$$P(y_{i,k} = 1 | x_{i,k}, s_{i,k})$$

- A recursive function of all previously observed events

$$s_{i,k} = g(x_{i,k}, s_{i-1,k})$$

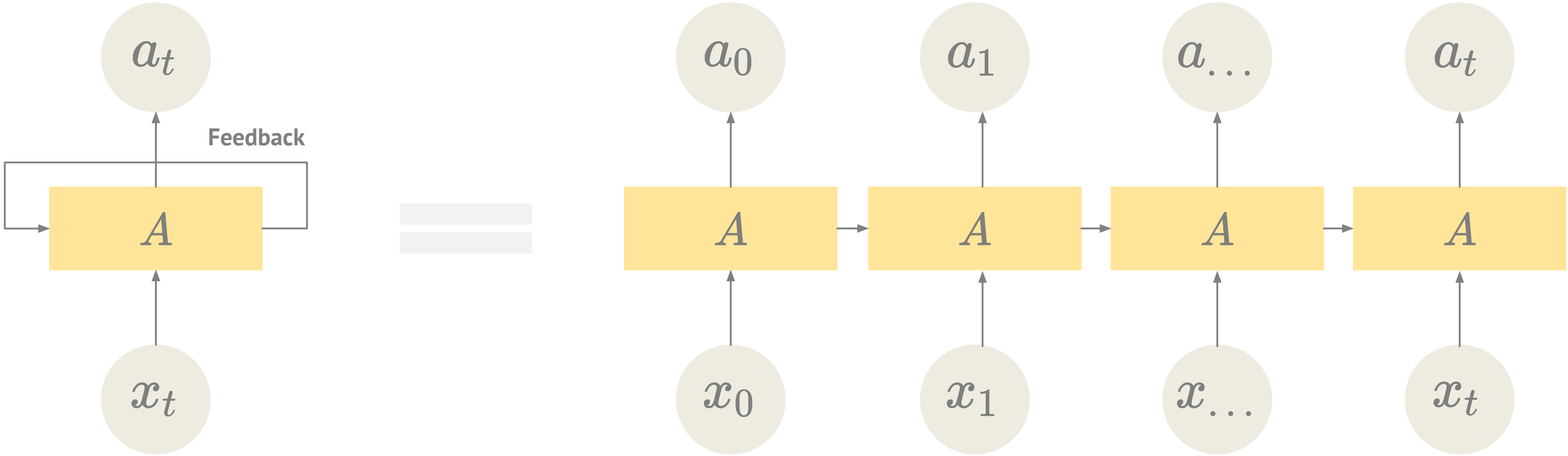
- If there are no previous events, we can assume the state is a vector of zeros.

Built-in feedback loops

- Recurrent networks include cycles for processing sequences
- Representing the influence of the present value of a variable in its future label

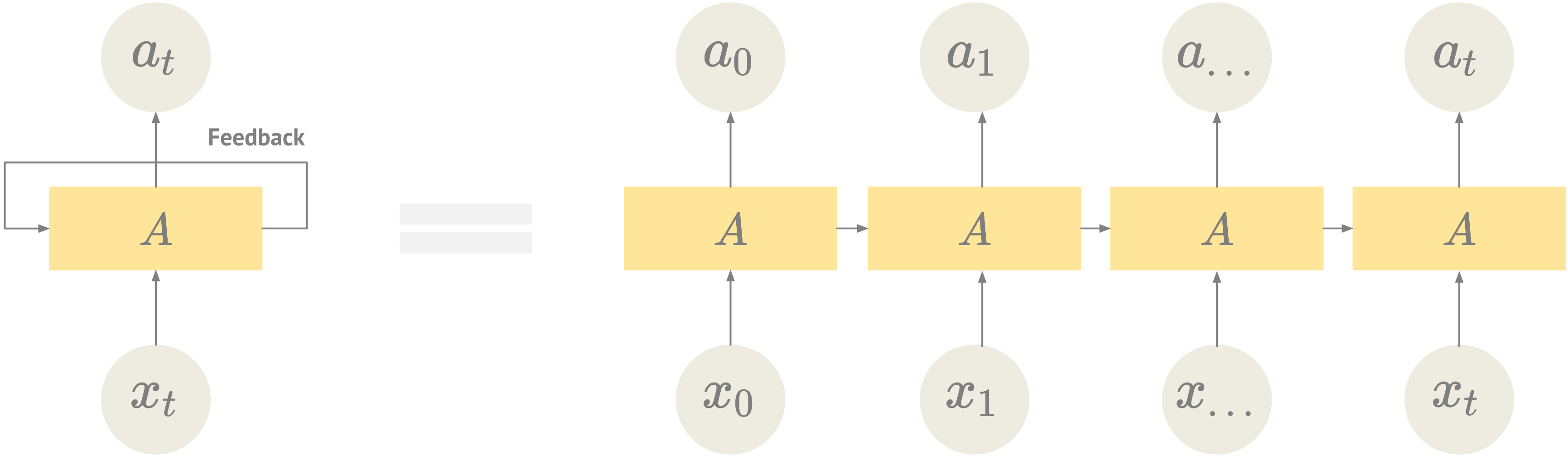
Built-in feedback loops

- Recurrent networks include cycles for processing sequences
- Representing the influence of the present value of a variable in its future label



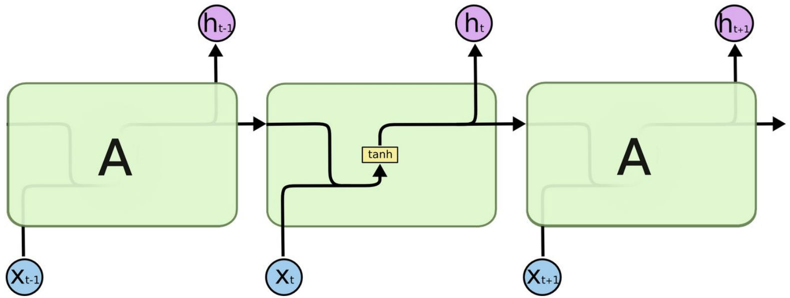
Built-in feedback loops

- Recurrent networks include cycles for processing sequences
- Representing the influence of the present value of a variable in its future label



- As multiple copies of the same network, sharing information sequentially.

Zooming-in on RNN memory cells

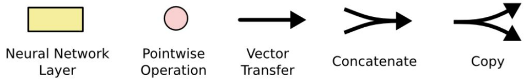


- Chains of repeating modules
- Standard RNNs have a simple structure
- Typically, with a single $\tanh$ layer

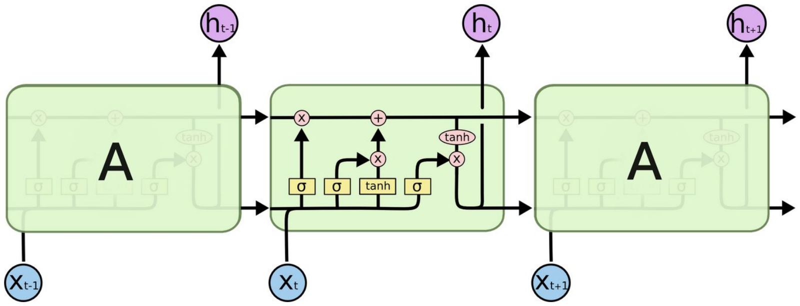
$$s_{i,k} = \tanh(W_s[x_{i,k}, s_{i-1,k}] + b_s)$$

- Struggle with long term dependencies.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

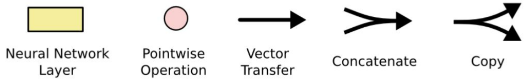


LSTMs for long-term memory

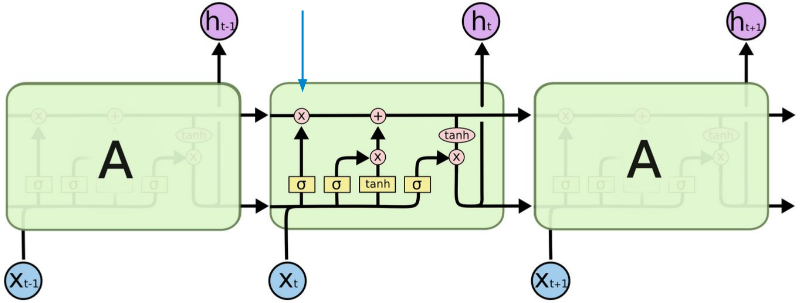


- Also chains of repeating modules
- Four interacting layers, instead of one
- The key is the cell state C at the top
- State goes through interactions (gates)
- Very easy to remain unchanged

http://colah.github.io/posts/2015-08-Understanding-LSTMs/



LSTMs for long-term memory



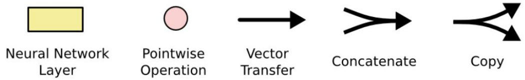
Forget gate layer

- Discard information from current state

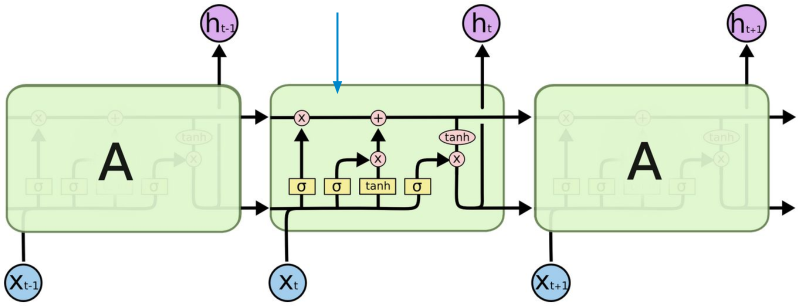
$$g^{(1)} = \sigma(W^{(1)}[x_{i,k}, s_{i-1,k}] + b^{(1)})$$

- Between zero (forget) and one (keep)
- For each element in the cell-state

http://colah.github.io/posts/2015-08-Understanding-LSTMs/



LSTMs for long-term memory



Input gate layer

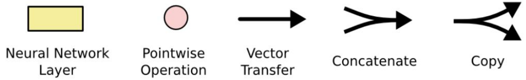
- Again, what state values to update

$$g^{(2)} = \sigma(W^{(2)}[x_{i,k}, s_{i-1,k}] + b^{(2)})$$

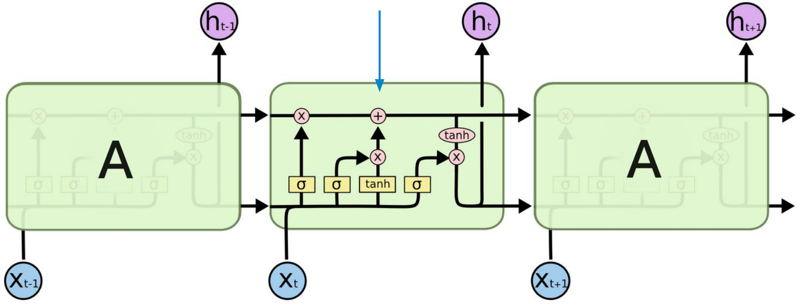
- Generate new candidate state values

$$C'_{i,k} = g^{(3)} = \tanh(W^{(3)}[x_{i,k}, s_{i-1,k}] + b^{(3)})$$

http://colah.github.io/posts/2015-08-Understanding-LSTMs/



LSTMs for long-term memory

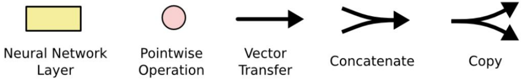


Updating the current state, C

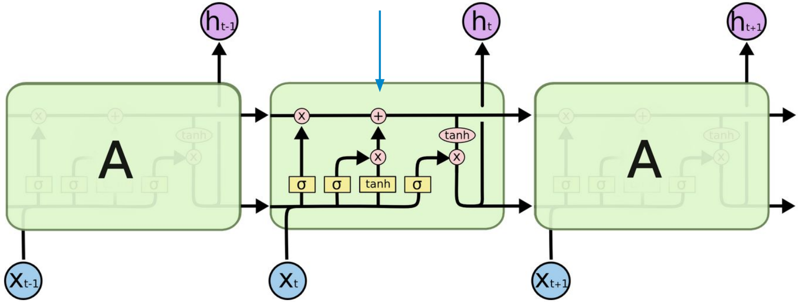
- Forget past information, add new one

$$C_{i,k} = g^{(1)} * Ci - 1 + g^{(2)} * C'_{i,k}$$

http://colah.github.io/posts/2015-08-Understanding-LSTMs/



LSTMs for long-term memory



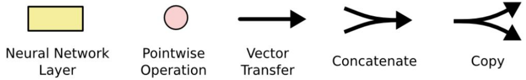
Output gate layer

- Filtered version of the current state

$$o_{i,k} = g^{(4)} = \sigma(W^{(4)}[x_{i,k}, s_{i-1,k}] + b^{(4)})$$

- $\tanh$ to push state to interval -1, 1

$$s_{i,k} = o_{i,k} * \tanh(C_{i,k})$$



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

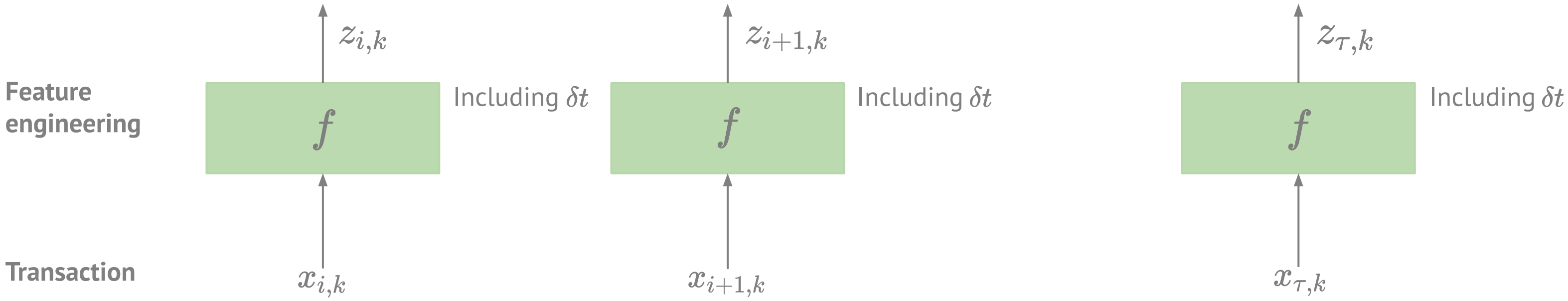
Problem Solution

Essential building blocks

Transaction $x_{i,k}$ $x_{i+1,k}$ $x_{T,k}$

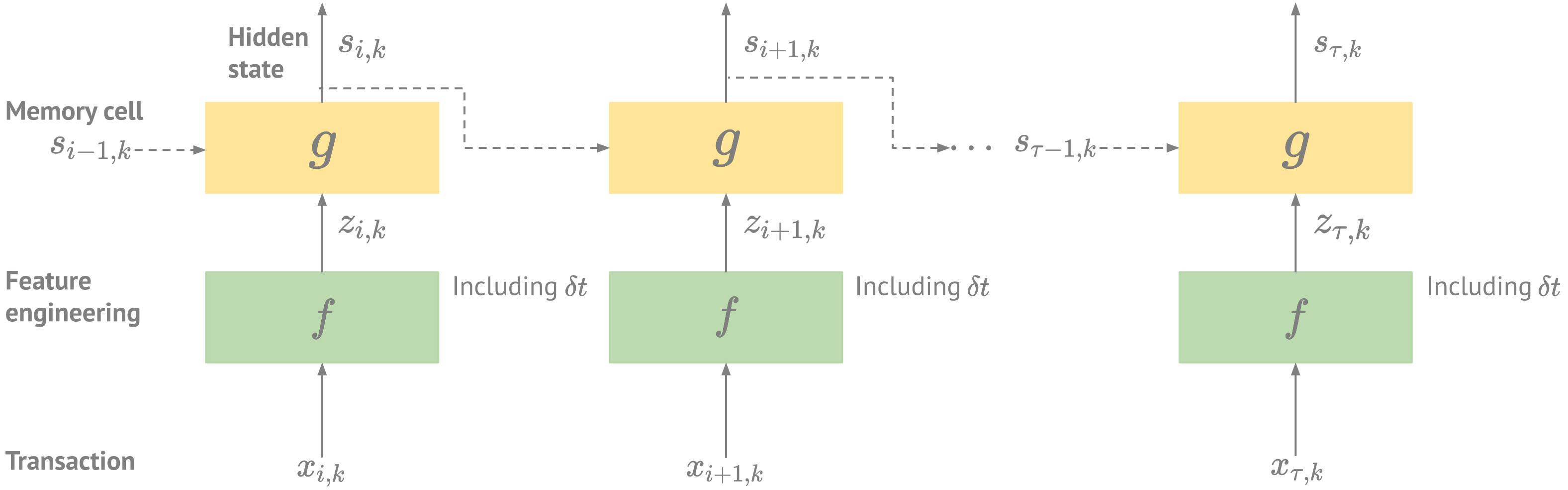
Essential building blocks

$$z_{i,k} = f(x_{i,k}) \quad (1)$$



Essential building blocks

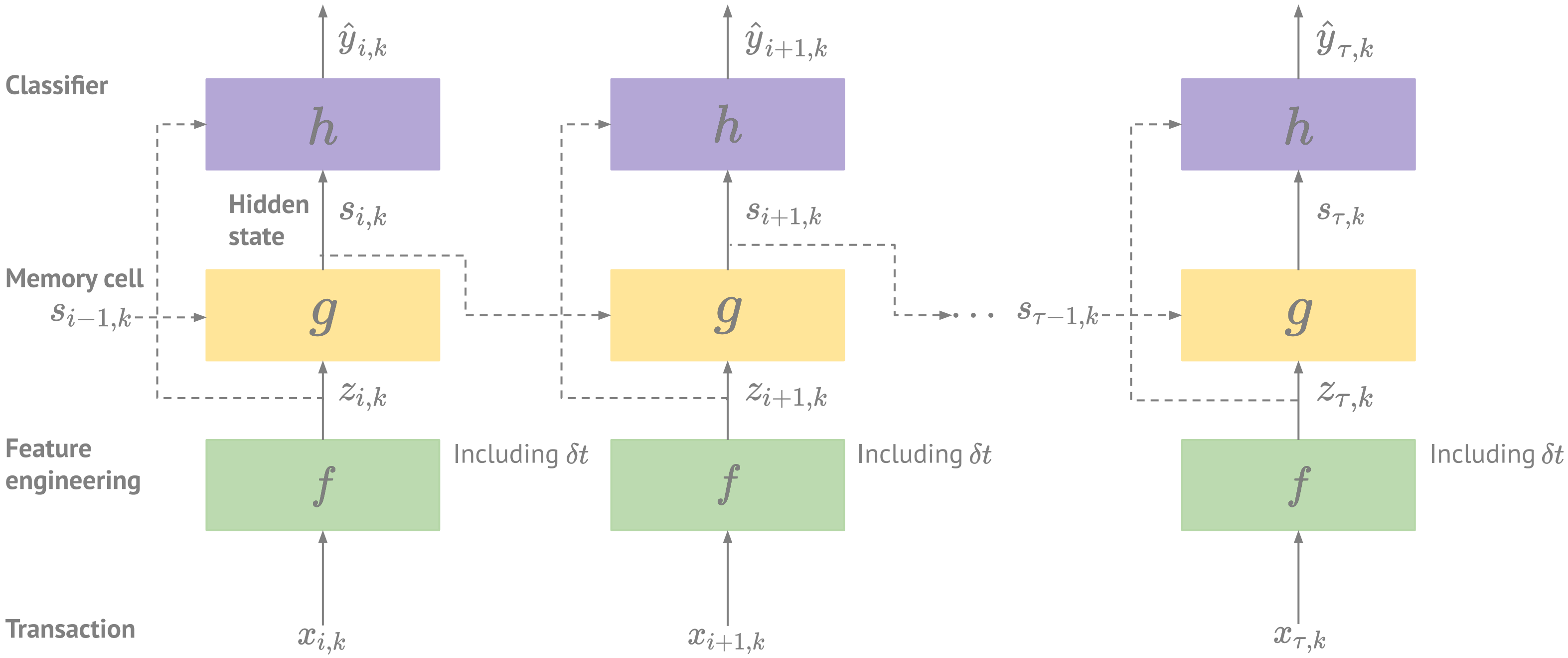
$$z_{i,k} = f(x_{i,k}) \quad (1)$$
$$s_{i,k} = g(z_{i,k}, s_{i-1,k}) \quad (2)$$



feedzai

Essential building blocks

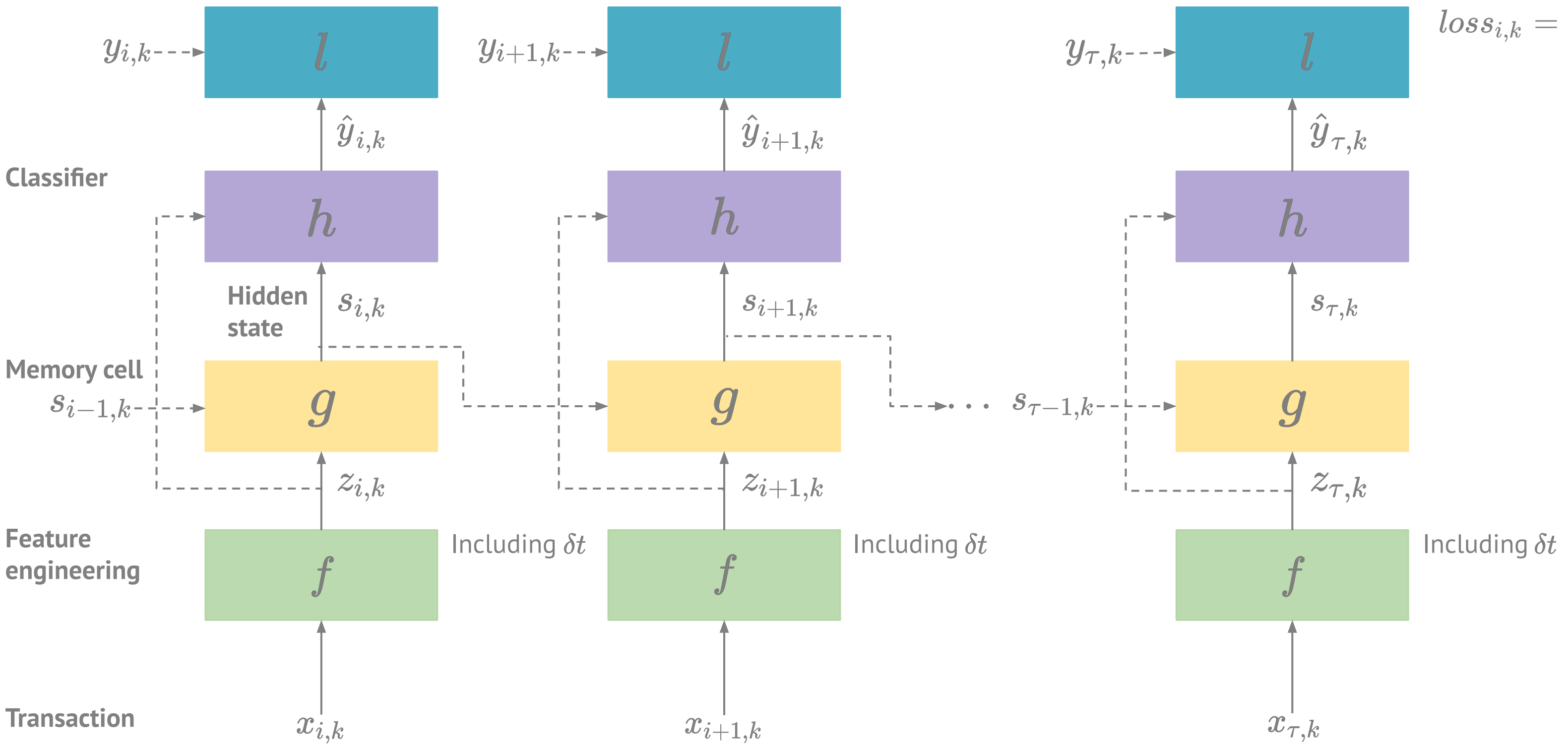
$$z_{i,k} = f(x_{i,k}) \quad (1)$$
$$s_{i,k} = g(z_{i,k}, s_{i-1,k}) \quad (2)$$
$$\hat{y}_{i,k} = h(z_{i,k}, s_{i,k}) \quad (3)$$



feedzai

Essential building blocks

$$z_{i,k} = f(x_{i,k}) \quad (1)$$
$$s_{i,k} = g(z_{i,k}, s_{i-1,k}) \quad (2)$$
$$\hat{y}_{i,k} = h(z_{i,k}, s_{i,k}) \quad (3)$$
$$loss_{i,k} = l(\hat{y}_{i,k}, y_{i,k}) \quad (4)$$



See it in action

https://medium.com/feedzaitech/casting-deep-nets-on-financial-crime-502617cab40b

Training

- Each sequence of transactions of a card is fed to the model
- Typically, we operate on minibatches of such sequences
- With different sequence length τ for each member of the minibatch

Training

- Each sequence of transactions of a card is fed to the model
- Typically, we operate on minibatches of such sequences
- With different sequence length τ for each member of the minibatch
- Once we have the loss for each transaction we adjust the model through SGD

feedzai

Training

- Each sequence of transactions of a card is fed to the model
- Typically, we operate on minibatches of such sequences
- With different sequence length τ for each member of the minibatch
- Once we have the loss for each transaction we adjust the model through SGD
- Repeat with a different minibatch of cards
- See the model improve over time

Takeaways

A few things to keep in mind

- Deep learning used on tabular data, perhaps the most common type of data
- Lengthy feature engineering to convey past information can be replaced with recurrent neural networks, this a kind of magic
- No free lunch theorem, “algorithms are equivalent when their performance is averaged across all possible problems”
- Pick an algorithm whose assumptions fit your particular problem
- Prefer algorithms that are well tested and properly researched.

Practical Resources

Practical Resources