Fraud detection without feature engineering
Pamela Vagata
Stripe

A global technology company that builds economic infrastructure for the internet

Help more companies get started and thrive, and ultimately grow the GDP of the internet
Fraud Prevention
Classical Machine Learning

- Easy to understand modelling techniques
- Models-as-Data
  - ML Production Systems can be runtime independent from training/offline experimentation systems
- Feature Engineering
  - Increasingly complicated features
  - Manual, arduous process
  - Complex infrastructure: aggregates, joins
    - ~consistency across online/offline
Learning Behavioral Patterns

- Fraudulent intent is latent in the observable behavioral patterns
- Model Engineering rather than feature engineering:
  - Simplify the feature complexity while improving prediction accuracy
  - Moving the toil of feature engineering from the ML engineer to the model itself
- Can we engineer a model that learns to predict fraudulent intent from raw behavioral sequences?
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data

```
RNN
```

```
User Signup
```

```
RNN
```

```
pr(fraud)
```
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data

\[ \text{RNN} \xrightarrow{\text{User Signup}} \text{RNN} \xrightarrow{\text{Verify Account}} p_r(\text{fraud}) \]
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data

\[ \text{User Signup} \rightarrow \text{Verify Account} \rightarrow \text{Account Details} \rightarrow \text{Logout} \]

\[ pr(\text{fraud}) \]
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data

![Diagram of RNN process]

\[ \text{RNN} \Rightarrow \text{User Signup} \Rightarrow \text{Verify Account} \Rightarrow \text{Account Details} \Rightarrow \text{Logout} \Rightarrow \text{Login} \Rightarrow p(fraud) \]
Recurrent Neural Networks

- Family of Neural Networks
  - Recurrent Loop
  - Models Sequential Data

```
RNN = RNN
  User Signup → RNN → Verify Account → RNN → Account Details → RNN → Logout → RNN → Login → RNN → Process Payment → pr(fraud)
```
Different Labels, Different Flavors

Many-to-one

Many-to-many
Encoding Event Sequences

- Encode event types as categorical values
- Timestamps: delta-encoded
- Categorical metadata:
  - Map distinct categorical values to a vocabulary
  - Jointly trained embeddings for categorical values
- Numeric metadata:
  - Scaled between 0 and 1

EVENT 0

- event_type: “signup”
- timestamp: “2017-12-30 00:00:00”
- event_metadata_categorical
- event_metadata_numeric
Training with Events

- Data Pipeline: leverage Spark to produce training data
- Serialized to parquet
- Sequence of events where each event is
  - Array of categorical\_vocabulary\_index
  - Array of numeric values
- Labels
  - For each event (if available)
  - For each sequence
- Deserialized with pyarrow library
  - Handles complex datatypes such as arrays
Model Architecture

Event Sequence → Embeddings → Projection

categorical
Model Architecture

Event Sequence

- Embeddings: categorical
- Projection: numeric

concatenate
Model Architecture

- Event Sequence
- Embeddings
- Projection
- Concatenate
- RNN
- \( pr(fraud) \)

Connections:
- Categorical
- Numeric
Model Architecture

Event Sequence → Embeddings
  ▼
  ▼ categorical

Event Sequence → Projection
  ▼ numeric

Event Sequence → concatenate

concatenate → RNN

RNN → pr(fraud)

pr(fraud) → Loss

Loss → Label
Results

ROC curve

TPR vs FPR for RNN and Production model.
Some Future Work

● Explanation models
● Applying Attention
● Applications to other modeling challenges at Stripe
  ○ \( f(\text{event\_sequence}) \Rightarrow \Pr(\text{label}) \)
● Data/Models that don’t fit in memory
  ○ Distributed/parallel training/serving
● Future challenges span:
  ○ Systems/infrastructure work
  ○ ML/modelling work
Key takeaways

- Combine deep learning and minimally transformed metadata from real-world events
  - Leverage computationally powerful hardware
  - Shifting compute from data-engines to modeling hardware
  - Reduce the complexity of feature-transformation data pipelines
- Mindset shift
  - From: combining features into increasingly complex features
  - To: raw signals that capture highest information related to the ML task
- Model Engineering
  - Design a model architecture for the ML task and available data

Thank you!