

Overton

Apple-flavored ML

2019/10/07 Nantes Machine Learning Meetup

Overview

Overview

What / When / Where

Publishing date

Early September

Conference

NeurIPS

Goal

ML software lifecycle management

Maturity

Production

Overview

Challenges to overcome

- Precise monitoring
- Complex pipelines
- Efficient feedback loop

Overview

Architectural choices

- Code-less deep learning
- Multi-task learning
- Weak supervision

Code-less deep
learning

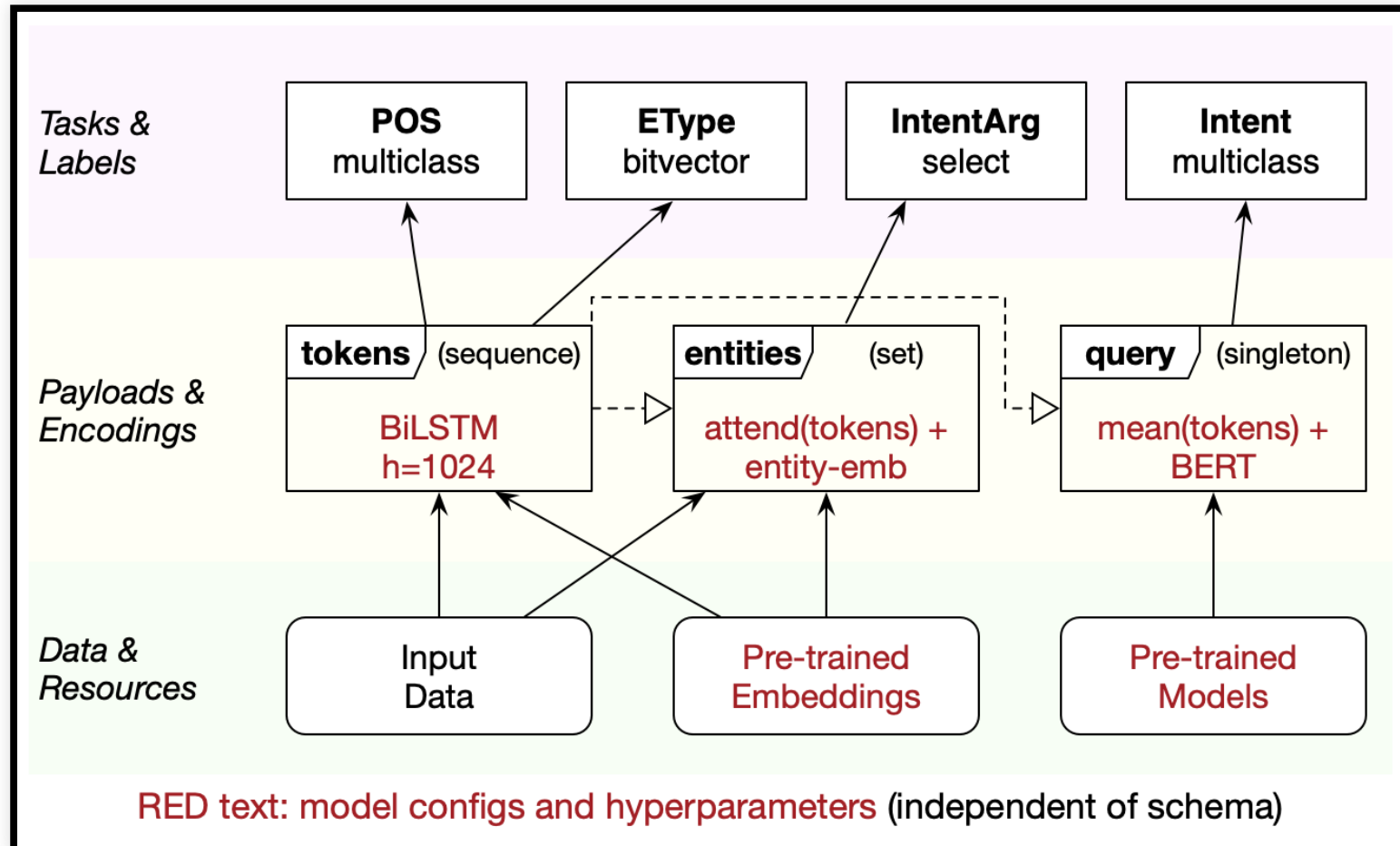
Code-less deep learning

Principles

- Models & training code = experts
- “black box” by engineers

Code-less deep learning

Modularity



Code-less deep learning Configuration

Schema

```
{
  "payloads": {
    "tokens": {
      "type": "sequence",
      "max_length": 16
    },
    "query": {
      "type": "singleton",
      "base": ["tokens"]
    },
    "entities": {
      "type": "set",
      "range": "tokens"
    }
  },
  "tasks": {
    "POS": {
      "payload": "tokens",
      "type": "multiclass"
    },
    "EntityType": {
      "payload": "tokens",
      "type": "bitvector"
    },
    "Intent": {
      "payload": "query",
      "type": "multiclass"
    },
    "IntentArg": {
      "payload": "entities",
      "type": "select"
    }
  }
}
```

Example Data Record

```
{
  "payloads": {
    "tokens": ["How", "tall", ...],
    "query": "How tall is the president of the  
united states",
    "entities": {
      0: {"id": "President_(title)", range: [4, 5]},
      1: {"id": "United_States", range: [6, 9]},
      2: {"id": "U.S._state", range: [8, 9]},
      ...
    }
  },
  "tasks": {
    "POS": {
      "spacy": ["ADV", "ADJ", "VERB", ...]
    },
    "EntityType": {
      "eproj": [[], ..., ["location", "country"]]
    },
    "Intent": {
      "weak1": "President",
      "weak2": "Height",
      "crowd": "Height"
    },
    "IntentArg": {
      "weak1": 2,
      "weak2": 0,
      "crowd": 1
    }
  }
}
```



Model Tuning

```
{
  "tokens": {
    "embedding": [
      "GLOV-300",
      "BERT",
      "XLNet"
    ],
    "encoder": [
      "LSTM",
      "BERT",
      "XLNet"
    ],
    "size": [
      256, 768, 1024
    ]
  },
  "query": {
    "agg": [
      "max", "mean"
    ]
  },
  "entities": {
    "embedding": [
      "wiki-256",
      "combo-512",
    ],
    "attention": [
      "128x4", "256x8"
    ]
  }
}
```



Fine grained ML
& multi-tasks

Fine grained ML & multi-tasks

Problems

- Lots of subtasks (implicit & explicit)
- Need to evaluate & monitor them
- Need to improve on them

Fine grained ML & multi-tasks

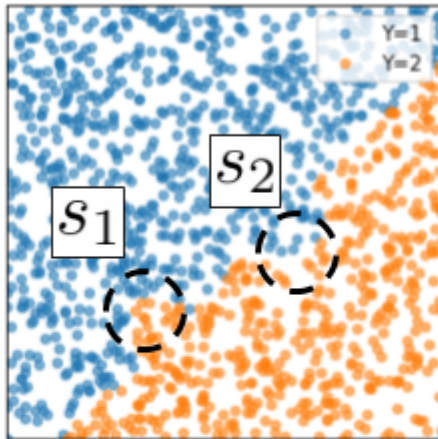
Approach

Slice-based learning

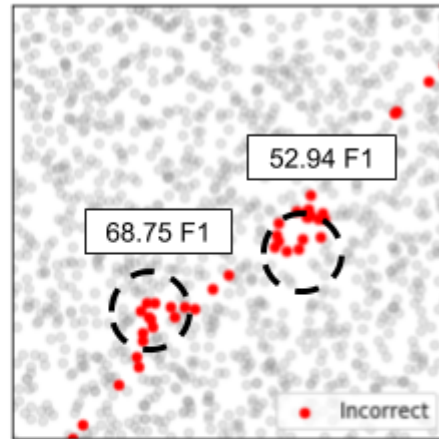
- Definition of data subsets
- Augmentation of model capacity
- Dedicated metrics

Fine grained ML & multi-tasks

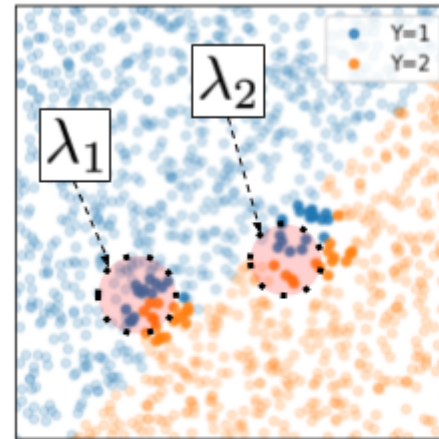
Slice-based learning



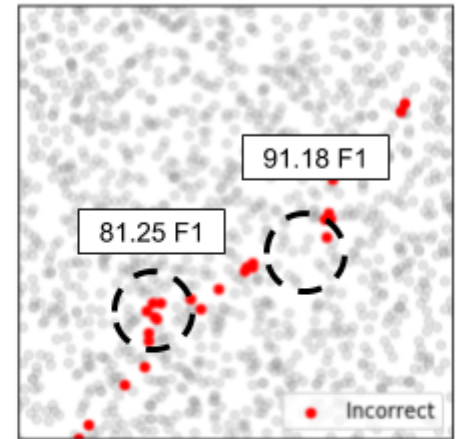
(a) Synthetic data with critical slices (dashed)



(b) Vanilla model errors



(c) User heuristically targets slices SFs (dotted)



(d) Slice-aware model errors

Fine grained ML & multi-tasks

Definition of critical data subsets

With “slice functions”:

```
def sf_bike(x):  
    return "bike" in object_detector(x)  
  
def sf_night(x):  
    return avg(X.pixels.intensity) < 0.3
```

Fine grained ML & multi-tasks

Slice experts

For each slice, an expert:

- should add capacity to the model
- has to know when to trigger
- has to have dedicated metrics

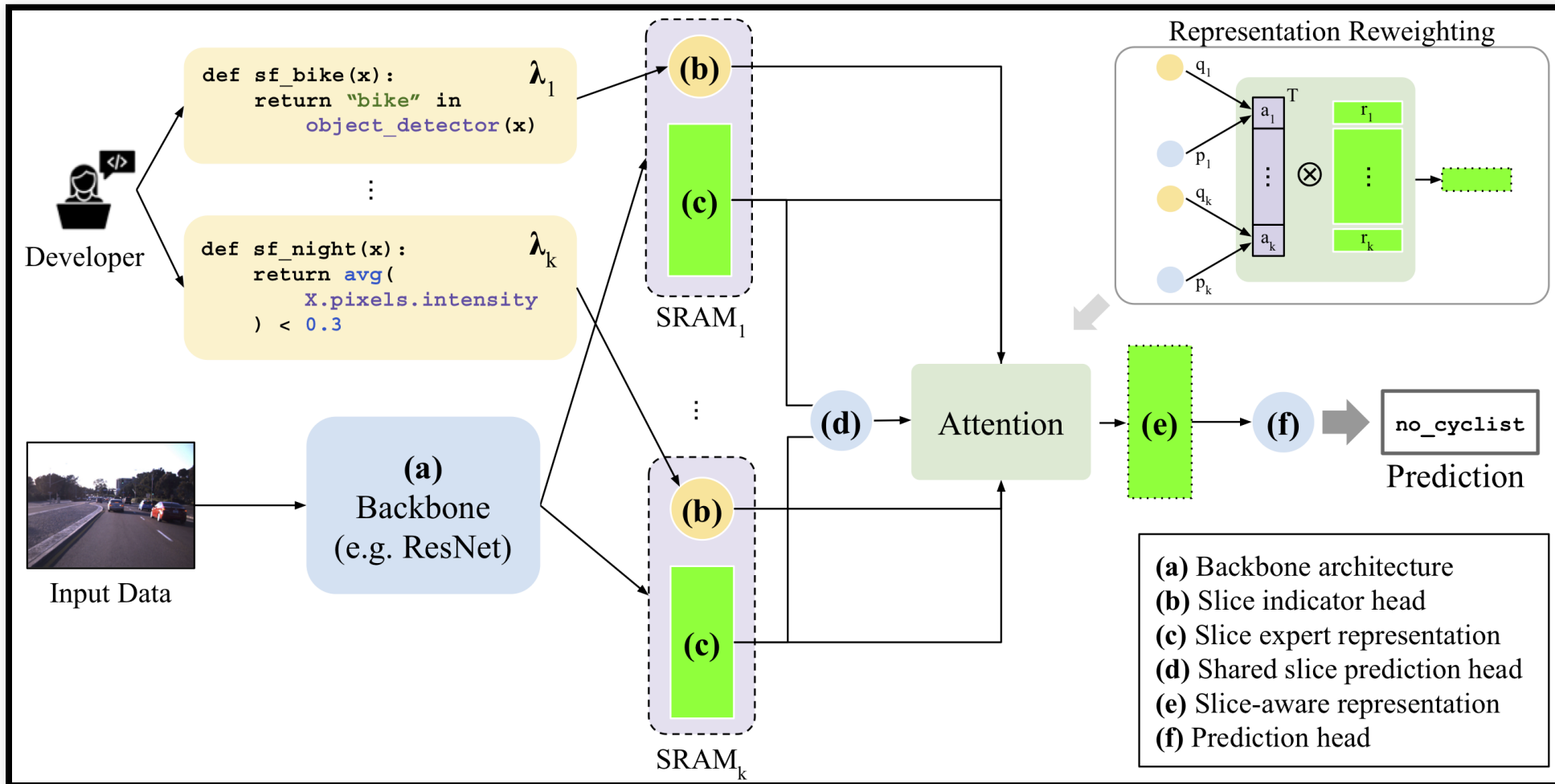
Fine grained ML & multi-tasks

Hard parts

- Noise : slices are defined with heuristics
- Scale 1 : when the number of slices goes up, does the model still run fast enough?
- Scale 2 : when the number of slices goes up, is the model still good enough?

Fine grained ML & multi-tasks

Proposed solution



Resources

- Snorkel blog “Slice-based Learning”
- NeurIPS paper “Slice-based Learning: A Programming Model for Residual Learning in Critical Data Slices”

Weak
supervision

Weak supervision

Problem statement

We need data, but:

- It's expensive, not available, yadda yadda
- Especially for interesting corner cases
- It's noisy

Weak supervision

Solution

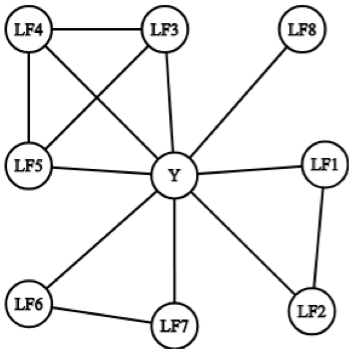
Use heuristics

```
@labeling_function()  
def lf_regex_check_out(x):  
    """Spam comments say 'check out my video', 'check it out', et  
    return SPAM if re.search(r"check.*out", x.text, flags=re.I) e
```

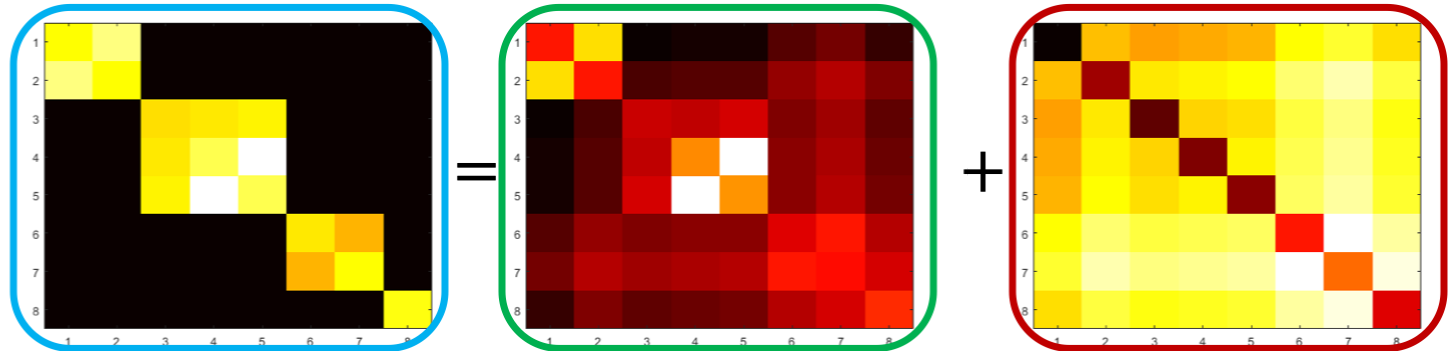
Weak supervision

Solution

Then correct the loss accounting for their correlation



$$(\Sigma^{-1})_o = \Sigma_o^{-1} + zz^T$$



Weak supervision

Implementation

Overton uses a modified version of the “Label Model”
from Snorkel.

Resources

- Snorkel blog « Introducing the New Snorkel »
- ICML paper “Learning Dependency Structures for Weak Supervision Models”

Conclusion

- Modularity (alike AllenNLP, tensor2tensor)
- Dedicated metrics & multi-tasks
- Improvements on critical data subsets
- Artificial data with theoretical corrections

Questions / Discussion