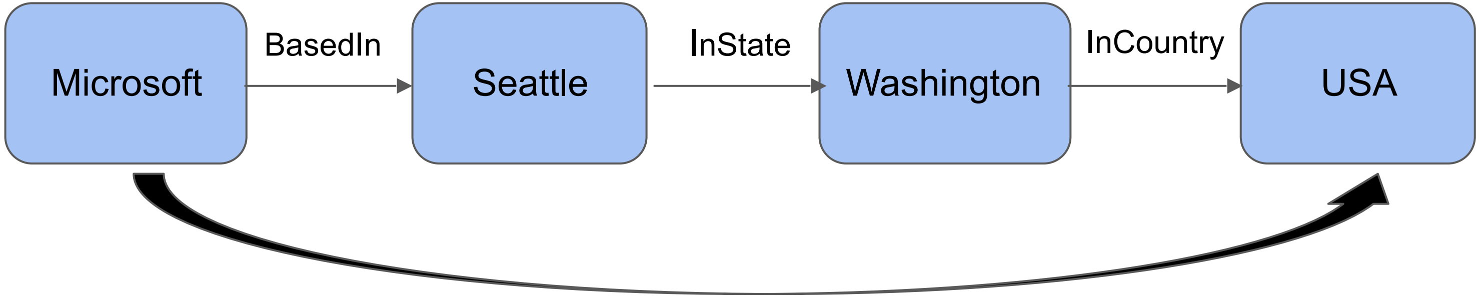
Reimplementing Neural Tensor Networks for Knowledge Base Completion in the TensorFlow framework

Dustin Doss, Alex Lenail, Clare Liu

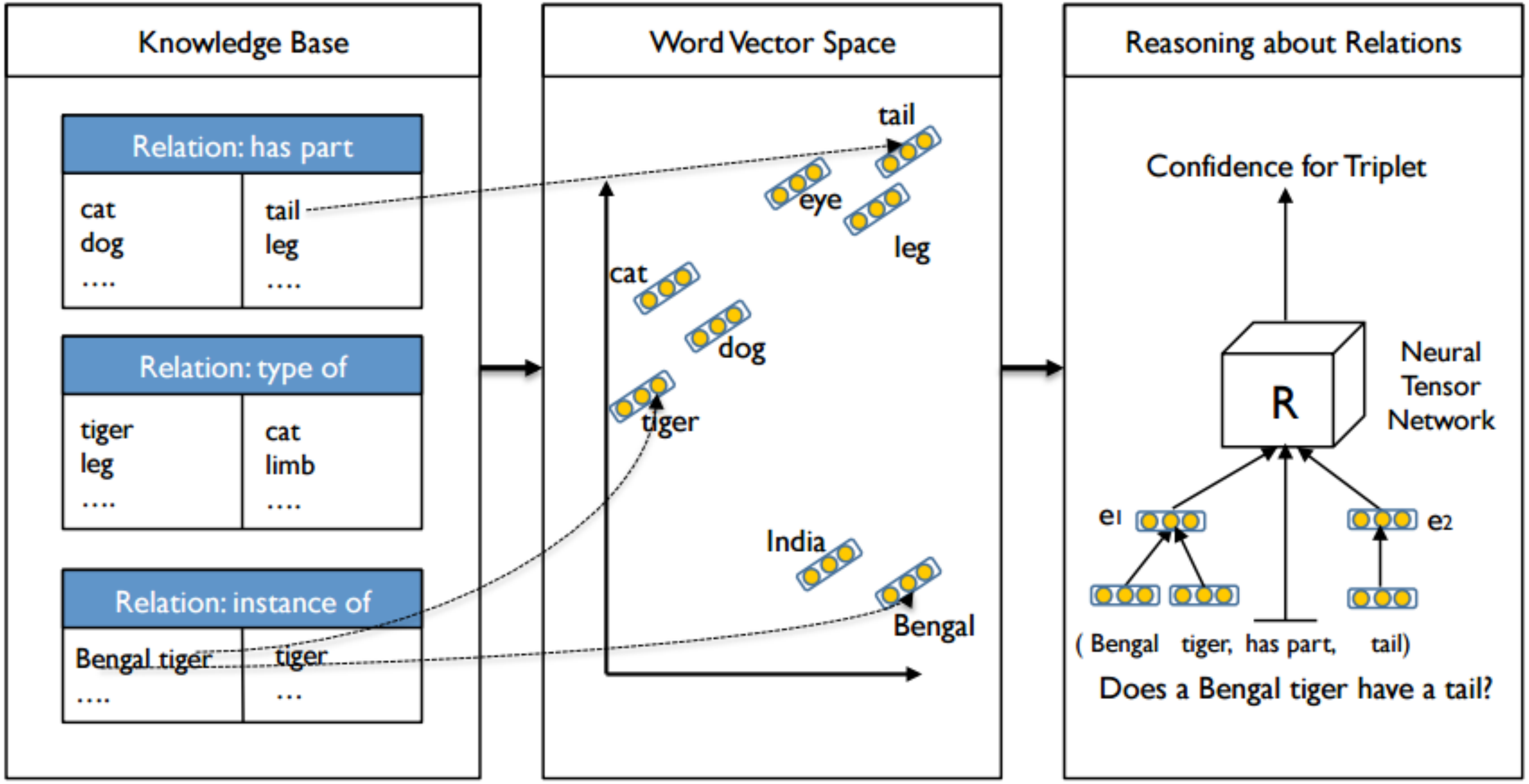
Introduction

- A **knowledge base** stores **entities** and the **relationships** between them to represent **facts about the world**.
- Knowledge bases suffer from **incompleteness** and an inability to reason over the relationships between entities
- We used **TensorFlow**, an open-source machine learning library that was recently released by Google, to develop a Neural Tensor Network (NTN) to **infer new relationships** between entities

Knowledge Base Completion



How can we infer that Microsoft is located in the USA?



Knowledge Base
Relation: has part
cat
dog
....
tail
leg
....
Relation: type of
tiger
leg
....
cat
limb
....
Relation: instance of
Bengal tiger
....
tiger
...
Word Vector Space
tail
eye
leg
cat
dog
tiger
India
Bengal
Reasoning about Relations
Confidence for Triplet
R
Neural
Tensor
Network
e1
e2
(Bengal
tiger,
has part,
tail)
Does a Bengal tiger have a tail?

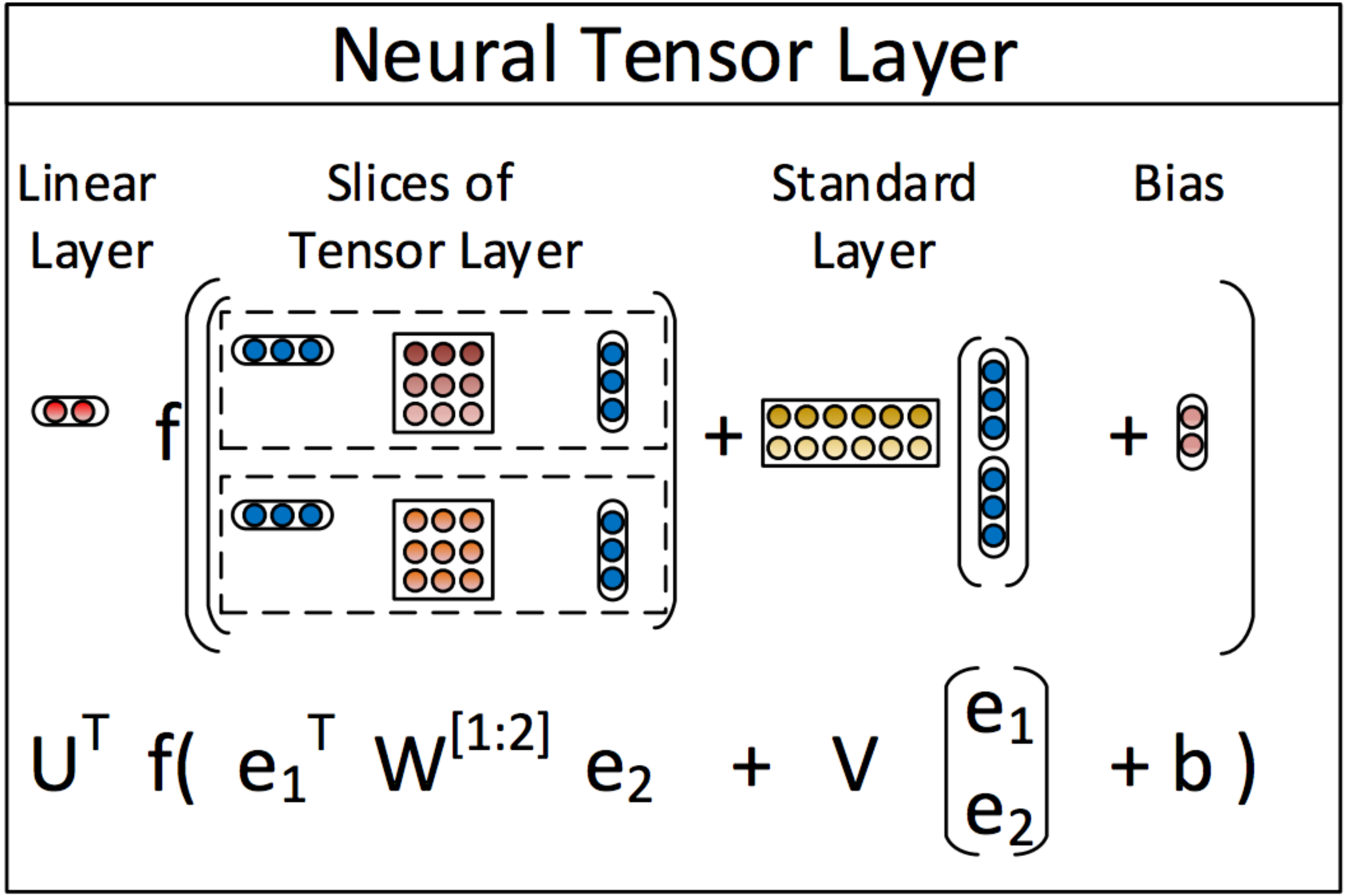
Neural Tensor Network

$$g(e_1, R, e_2) = u_R^T f\left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R\right)$$

g = score for the likelihood that *e1* and *e2* are in the relationship *R*

W = tensor with *k* slices
f = tanh activation function

b = bias
U, *V* = standard neural network parameters


Neural Tensor Layer
Linear Layer
Slices of Tensor Layer
Standard Layer
Bias
f
+
+
$U^T f(e_1^T W^{[1:2]} e_2 + V \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} + b)$

TensorFlow

TensorFlow

TensorFlow is a open-source library released by Google for **numerical computation** in data flow **graphs**

Represents algorithms as directed acyclic graphs (DAGs), nodes as operations, and edges as schemas for tensors

Datasets

1. **Wordnet** - a knowledge base for the English language, which groups words into sets of synonyms and contains relations between these sets. Wordnet contains 38,696 entities and 11 relations.

2. **Freebase** - a collaborative knowledge base which connects entities as a graph. Freebase contains 75,043 entities and 13 relations

Inputs and Outputs

Inputs: Entity-relationship triplets of the form (e1, R, e2), where e1 is the subject, e2 is the object, and R is the relation

Outputs: Prediction = 1 if e1 and e2 are in relationship, -1 otherwise

Entities:

male
united_states
female
politician
germany
writer
united_kingdom
england
france
paris
new_york
actor
london
lawyer
italy
catholicism

Relations:

gender
nationality
profession
place_of_death
place_of_birth
location
institution
cause_of_death
religion
parents
children
ethnicity
spouse

Training set:

antoine_brutus_menier	religion	roman_catholic_church
denys_rayner	cause_of_death	cancer
nietzchka_keene	place_of_death	madison
friedrich_bessel	profession	mathematician
thomas_harrison_1740	profession	engineer
richard_brautigan	profession	novelist
anthony_asquith	location	london
robert_noyce	profession	physicist
ignaz_franz_castelli	profession	dramatist
russell_bufalino	gender	male
mary_de_bohun	nationality	kingdom_of_england
cato_maximilian_guldberg	place_of_death	oslo

Evaluation

Corrupt a selection of entity-relation triplets by randomly switching entities between correct triplets

Evaluate how many of these triplets are classified correctly

(Pablo Picaso, nationality, Spain)
(Barack Obama, nationality, USA)

corruption →

(Pablo Picaso, nationality, USA)
(Barack Obama, nationality, Spain)

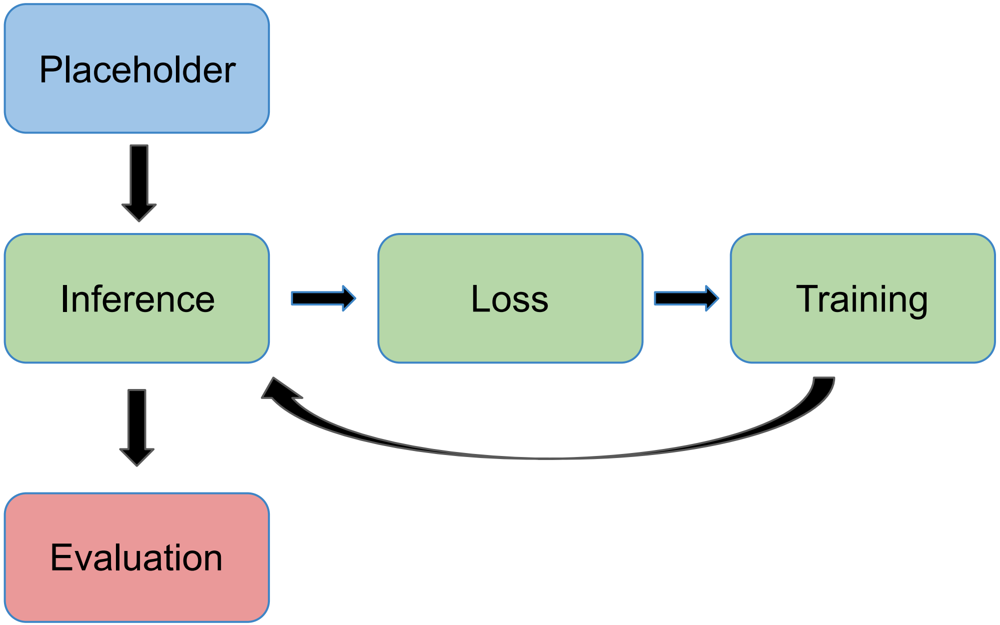
Can we predict (Pablo Picaso, nationality, Spain) and (Barack Obama, nationality, USA) again?

Data flow

Initialization

Building Graph

Evaluation



Loss Function

$$J(\boldsymbol{\Omega}) = \sum_{i=1}^{N} \sum_{c=1}^{C} \max\left(0, 1 - g\left(T^{(i)}\right) + g\left(T_c^{(i)}\right)\right) + \lambda \|\boldsymbol{\Omega}\|_2^2,$$

- $T^{(i)}$ is a correct relation triplet and $T_c^{(i)}$ is a corrupted relation triplet.
- We want to maximize the score of $T^{(i)}$ while minimizing the score of $T_c^{(i)}$
- Ω represents the model's parameters (*U*,*W*, *V*, *b*, *E*)

Baselines

Model	**Wordnet**	**Freebase**
Distance Model	68.3	61.0
Hadamard Model	80.0	68.8
Single Layer Model	76.0	85.3
Bilinear Model	84.1	87.7
Socher’s NTN	86.2	90.0

Remaining Questions & Future Work

- Adjoining tensor layers indexed by R into one big tensor?
- Varying k?
  Varying the embedding size?
- How much do the parameters move?
- Relations aren't necessarily so fixed and deterministic — could we represent relations as vectors as well?
- Is there some way to properly use RNNs for entity construction from words?
- Are there better ways to represent knowledge than knowledge bases?

Conclusions

- **Tensorflow is slow**! Multi-GPU support is on the way in a couple weeks.
- **Tensorflow is still quite buggy**! We had to ask them to fix a lot of stuff, and we ran into a nasty surprise just when we thought we would be done.
- **Tensorflow documentation needs work** Our entire algorithm hinged on an operation which turned out not to work the way most other operations do — which wasn't mentioned in the docs.
- **Tensorflow is magic**. Automatic Differentiation (AD) applies the chain rule on your operations meaning that once you define inference, training is trivial.