Vorlesung 1 Fragen:

- Was sind Nachteile von MA?

Black box- whats happening inside / hard to interpret

can only learn from the data

Unsupervised Learning:

- learn Probabilties/distributions

Supervised Learning:

- estimate function that predicts a label of given x

Definition von MA?

computer program is said to learn from experience E (data) with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E

What are discrete and continuous loss funcitons. Whats the difference:

Discrete loss : Acc, Recall, Precision Discrete loss functions cannot indicate how wrong a wrong decision is. Used for evaluation

Continuos loss: Squared error; differentiable. ==> use them for gradient descent

Used for optimization

Vorlesung 2 Fragen:

Hadamard product: Element-wise matrix multipication

$$||\boldsymbol{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$ L1/L2 Norm mit p=1 p=2

Bescheibe was eine loss function macht:

- L beschreibt einen positiven Wert. Je kleiner der Wert, desto besser

- bekommt ein predicted Label und ein tatsächliches Label und berechnet Wert.

- supervised Learning berechnet expected Loss.

Norm: Gibt eine Art Größe an. (Länge, max singulärwert, Maximum einer Funktion)

Vorlesung 3 Fragen:

- Was ist Linear Regression? Wie funktioniert es?

Einfachste art der Prediction von y gegeben x. mit einem geg. Parametervector.
X * w = prediction.

- w wird gelernt mit hilfe von MSE bzw der Normal Equation. MSE wird abgeleitet und gleich 0 gesetzt und nach w aufgelöst.

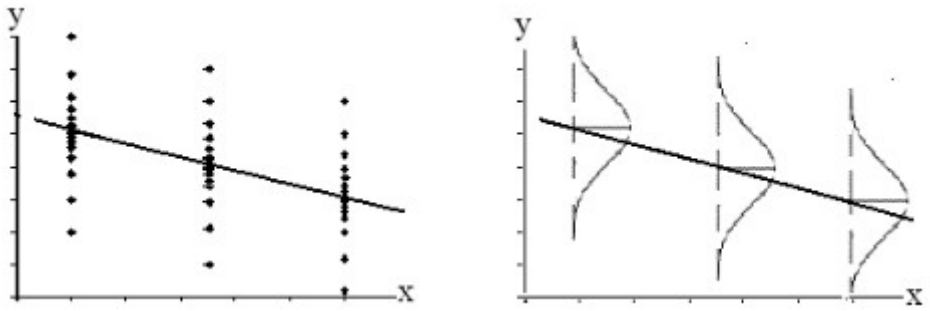
- Was ist maximum Likelihood? Wofür benutzt man sie

Man versucht die Wahrscheinlichkeitsdichte zu maximieren. Dabei wird der Gewichtsvektor getunt. Für supervised und unsupervised Learning

- argmax p(y|x,theta)

- argmax p(x,theta)

linear regression kann auch mit Maximum Likelihood berechnet werden. Anstatt y vorherzusagen, einfach die Wahrschienlichkeitsverteilung p(y|x) modellieren. Dabei kann x mehrere Values haben ==> Verteilung.



Cond. Neg log likilihood und MSE is equivalent μ=θTx

MSE kann probabilistisch gemacht werden mit Gaussian um das vorhergesagte y

==> Maximize Likelihood of training data (not bayesian)

==> MLE is more often used

Was ist Log. Regression. Was ist der Unterschied zu Linear Regression

- dient der Klassifikation

Logistic Regression modelliert mehrere Wk für y=0,1

lineare Feature-Gewicht kombination, gefolgt von Sigmoid funktion. Kann als WK für Label interpretiert werden. Wen >0,5 == label1

Vorlesung 4:

1. Optimization “loss”, cost, objective funct etc:

Was ist Optimization denn im Allgmeinen:

Wir minimieren bzw. Maximieren eine Funktion f(theta)
Convexe Funktionen werden damit minimiert, indem wir suchen, wo die Ableitung der func = 0 wird.

Bei Konvexen Funktionen schwieriger das globale minimum zu finden.
Finden auf jeden fall lokales minima mit dem gradienten

Formel für Gradient Descent

$\theta_{t+1} \leftarrow \theta_t - \eta \nabla_\theta J(\theta_t)$ konvergiert, wenn Gradient ~ 0

Was sind dabei die Probleme:

Kann verschiedene lokale Minima finden, abhängig davon, wie initialisiert ist und wie groß die Lernrate/Stepsize ist

Wie werden die Parameter geupdated:

Es werden alle parameter geupdated. Sigmoid liefert einen Wert zw. 1 und 0. Abhängig vom Label y wird dann das Gewicht erhöht oder verringert. Je näher sigma an tatsächlichem label, desto weniger Anpassung.

$$\boldsymbol{\theta}_{t+1} := \boldsymbol{\theta}_t + \eta \sum_{i=1}^{m} (y^{(i)} - \sigma(\boldsymbol{\theta}_t^T \boldsymbol{x}^{(i)}))\boldsymbol{x}^{(i)}$$

Welche arten der der Optimization kennen Sie:

(batch) Gradient Descent / Stochastic Gradient Descent

SGD uses random samples (even single examples). Gradient wird also annähernd berechnet.

2. Deep Feed Forward Networks

Warum benutzen wir Feed Forward Networks:

Haben keine Möglichkeit komplexe nicht lineare Probleme (XOR) mit einer Funktion darzustellen

Was ist der output eines FFN, wie bekommt man hin:

Meist eine WK oder eine Klassifikation. Softmax / Sigmoid zb.

Was ist charakteristisch für eine gute activation function:

liefert große Gradienten und hilft beim lernen. Sigmodial best for final layer. Relu for hidden Units, damit Gradient konst. Bleibt.

Was ist Forward/Backpropagation?

Forward ist der normale Durchlauf durch das NN. Für Input x mit Loss(theta) eine Ausgabe y bekommen.

Im backward schritt, wird der Gradient of cost berechnet für jedes Parameter/Layer berechnet. Die Parameter werden mit der chain rule geupdated.

Warum benutzt man Regularisierung? Welche Arten kennen Sie?

Wir wollen Modell mit niedrigen Gewichten (hohe Gewichte == Overfitting). L2 L1 Norm.

NEUER VORLESUNGS BLOCK:

Vorlesung 5

Warum Word embeddings?

Worte als Vektor, damit Programm es interpretieren kann. Jedes Wort bekommt eine Zahl. ==> Lookup in Vektor.

Dense WE sind trainierbare Word Vekoren. Man kann die Ähnlichkeit zw. Zwei WordVektoren berechnen.

Wie bekommt man Word Embeddings? (W=WordEmbeddingMatrix)

Pretrained oder selbst erstellte mit supervised Data: W random initalisieren, Als Input für RNN/CNN etc ==> Backpropagation modifies the Matrix accordingly. Ähnliche Worte ==> Ähnliche Vekotren

Was ist Skipgram? Wie funktioniert es?

Haben unsupervised training. Maximiere die Likelihood der Kontextwörter, gegeben das Center-Wort.

Was ist CBOW? Wei funktiniert es? Unterschied zu Skipgram?

Genau anders rum. Maximize Likelihood of center word given context words.

Naive Softmax model (V = to be predicted, w=given context/center)

- Skipgram: Softmax von V * w(center word)
- CBOW : Softmax von V * w(context_words)

w paradigmatic relations, v syntamatic relations.
==> Problem: müssen |V| dot-products rechnen. ==> slow

Hierarchical softmax model?

Machen eine Hierachische Struktur, in der Jeder Knoten ein lernbares Gewicht hat. Wir können P(w2|w1) berechnen, indem wir uns die Parents anschauen und wie Wk berechnen, ob das linke oder rechte Kind vorhergesagt wird. ==> Tiefe des Baums bestimmt Complexity.

log2(|V|) in binary Tree

Warum Negative sampling und was passiert dabei?

Anstatt alle Wk-Verteilung über das ganze Vokabular zu berechnen, berechnen wir die binary Wk für eine kleine Zahl von Wortpaaren ==> kein Language Model ==> Die Vorhersagen von Skipgram/Cbow interessieren und auch nicht, sondern nur die Wortvektoren.

- Für jedes beobachtete Wortpaar w werden x wordpaare mit random wörtern kreiert (w,v)

==> sage voraus, ob das Wortpaar exisitert oder erfunden wurde.

==> Gradientenupdate passt die Wortvektoren aneinander an.

Was ist Fasttext? Anwendungen?

Unbekannte Wörter haben keine Wortvektoren. (Subwords können helfen)
Nehmen char-ngrams als Wortvektoren auf. Und Definieren w neu.

Applications of pretrained word embeddings:

Frezing und fine tuning sind viable.
- Pro finetuning: learn features relevant to task.
- Freezing: training könnte overfitten.

Wir haben word Analogys: von Tokio ==> Warschau

Können übersetzen, wenn wir kleine Menge an übersetzungen haben. Lernen Übersetzungsfunktion und nehmen ähnlichstes wort

---

Vorlesung 6:

Was ist mit neural network achitecture gemeint?

CNN, RNN, Fully connected, Transformers: Der Aufbau der Verbindungen zw. Den Nodes eines NN

Was sind filter, Padding, stride, nonlinearities/bias im context von cnns?

- filter können sich eine inputsequence auf einmal anschauen und mit backpropagation trainiert werden. Sollen lernen welche Elemente der Sequenz wichtig und welche unwichtig sind. Die Dimensionen des Filters werden auch Kernel size genannt. Die Dimensionen des Input heißen Channels (WordEmbedding Dimensionality in NLP)

- Eine Inputsequence wird gepadded, wenn der Filter über sie hinausgehen sollte. Meist mit 0 oder mit einem average.

- stride ist die Schrittgröße mit der der Filter über den Input geht

- Meist wird ein Bias hinzugefügt, auf den eine nonlinearität angewendet wird (RELU)

Wie funktioniert Backpropacation in einem Covolution Layer?

Was sind Filterbanks?

Tensor an Filtern, die alle die gleiche Form haben. Jeder Filter wird einzeln/unabhängig auf den Input angewandt. Die Filterergebnisse werden dann wieder aufeinander gestackt. X1 x.. x XN xM mal F1 x ...x FN x M x J ergibt Tensor T1 x….xTN x J

Was sind ein paar Annahmen hinter convolution?

- Eine Information kann aufgrund der Filtersize nur begrenzt weit, weitergereicht werden (Locality)

- Die Filter sind immer die Gleichen. Es kann keine Varianz geben

- simple –> Complex: setzen aus kleinen Teilen größere Zusammen (bei mehereren CNNs)

Was ist Pooling? Pooling Layers?

Hat keine Parameter. Holt Average oder Maximum aus einem Grid des Inputs. Verkleinert Tensorsize. Fasst Input zusammen.

Wie wird Pooling/CNN in NLP benutzt?

CNNs mit Filterbanks. Only one convolutional layer. Not as deep as in Computervision.

Pooling selten genutzt. i.e. Filter bank. Max jeden Filter und damit sentiment prediction.

Vorlesung 7

Was sind RNNs?
Input wird Sequentiell verarbeitet.

Wie sieht die klassische RNN Zelle aus, was braucht man dafür? Wie funktioniert ein RNN?
Input (Embeddings), Initialzustand (hidden), Parameter(=Gewichte), Aktivationfunction (tanh).
Haben sequentiellen Input. Nach dem ersten Element ergibt sich h(1). H(2) wird in abhängigkeit zu h(1) berechnet. D.h. Jeder Input wird in abhängigkeit zu seinen Vorgängern berechnet. $\tanh(Wx_i + Vh_{i-1} + b)$

Multiplizieren Gewichtsmatrix V mit dem alten hiddenVector h_i-1. Multiplizieren W und x_i zusammen. Addieren die beiden mit b. Und berechnen tanh.

Wie funktioniert Backpropagation in RNNs?
Berechnen Loss und machen dot-Product mit dem Gradient der

Parameter/Gewichtsvektoren W und V. (Jeder Gradient von Layer (h_i) wird mit Loss multipliziert und geupdatet)

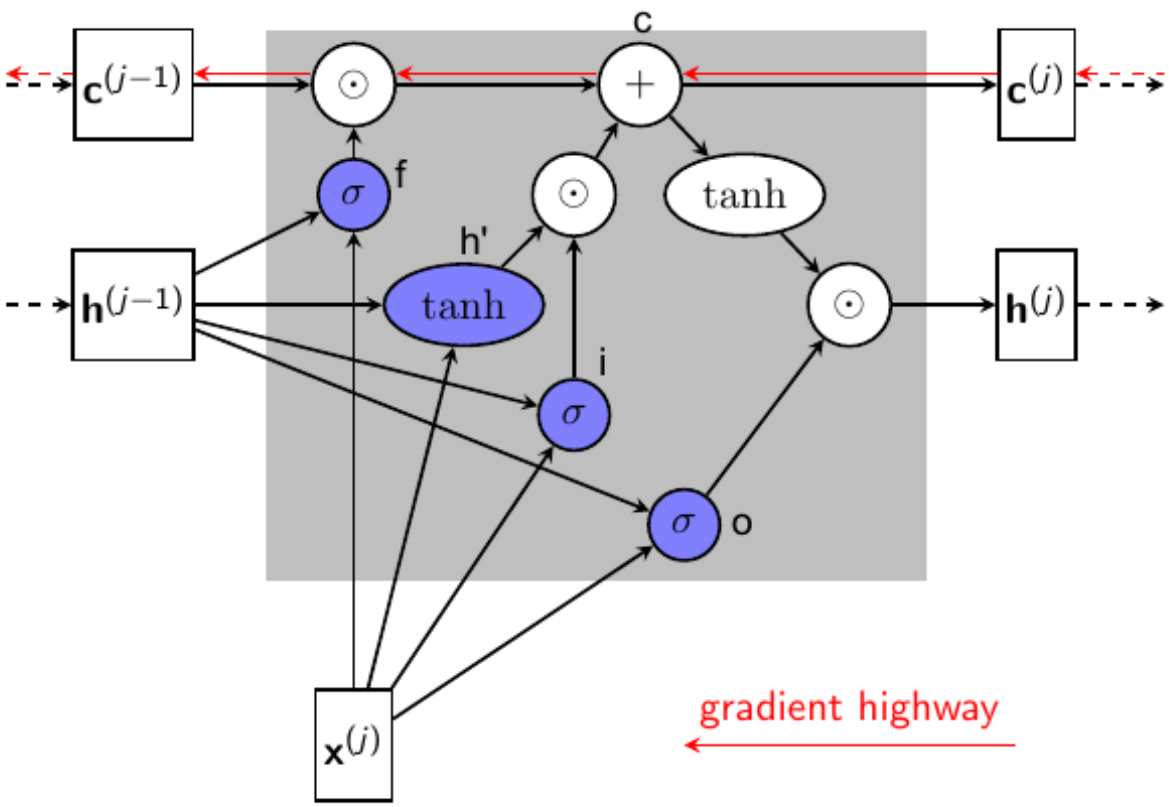
Erkläre das Problem vom Vanishing Gradiant

Älterer Input wird “vergessen. D.h. wenn der Input sehr lang ist, dann haben die ersten Eingaben weniger Einfluss auf den Loss vom Gradienten. ==> Schlecht, weil wichtige Informationen verloren gehen können. Die Gewichte werden nicht stark genug angepasst bzw. Garnicht.

LSTM erklären. GRU erklären. Was sind die Unterschiede/Gemeinsamkeiten?

LSTM: Ändert die Architektur der RNN Zelle. Haben Short-term und long-term memory.

Fügt 3 gates hinzu, welche entscheiden welche Information hinzugefügt/gelöscht werden sollen. (Forget, Input, Output, h’: Kandidaten, actual h)
o: Entscheidet was aus c in h kommt
I: Entscheidet was aus kandidaten in c hinzugefügt werden soll
f: Entscheidet was aus c gelöscht werden soll
c: für gradienten
i,f,o,h’ haben alle Parameter/Gewichte (theta)

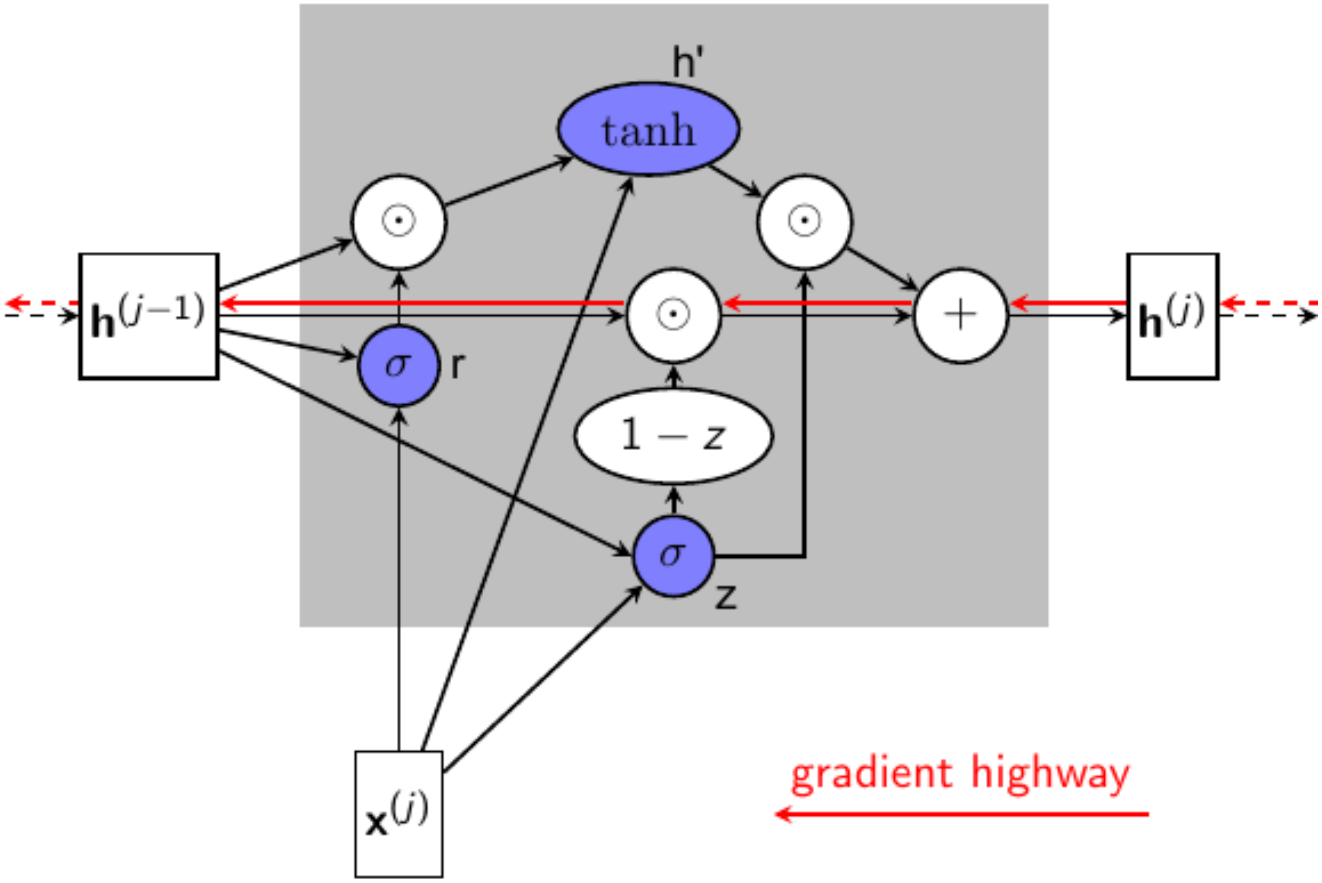


GRU: Gated Recurrent Unit

ein State und 3 Parametersets.

h beinhaltet short und longterm memory.

update-gate z, reset gate r, candidates h' haben alle eigene Parameter.



c fällt weg und Gradient kann weiterhin über h berechnet werden.

Was sind RNN Anwendungen?

Tagging, Autoregressive Language Modeling, Sequence2Seqence (translation)

Was ist autoregressiv Language Modeling:

feed-forward model which predicts future values from past values.

Was ist ein Multilayer RNN? Bidirectional RNN?

Multilayer sind mehrere Layer mit eigenen Parametern, die aufeinander gestackt werden. Bidirectional LSTMs laufen den Input von vorne und von hinten ab. (nicht für autoregressiv LM geeignet. Model würde zukünftige Inputs kennen, MT ähnlich. nur encoder darf bidirectional sein)

Was ist das Problem mit RNNs?

State h_j wird vermutlich ein Bottleneck, da es zu viele Informationen über vorherige Inputs halten muss. Lange Sequenzen können das Problem haben

Was macht Attention besser? Wie funktioniert Attention?

Attention hilft dem Decoder auf die ganze Sequenz zuzugreifen.

haben query, key und value Vektor. Scoring funktion a. Multiplizieren unsere Inputs mit Random q,k,v Matrix. Bekommen q,k,v Vektoren für jeden Input an position i .

q*k gibt scores. (Bei Vaswani. Bahdanau benutzt FFN) Über scores wird scoring function angewandt (z.b Softmax). Das Ergebnis wird der Alignment Score genannt und wird mit v multipliziert. Natürlich immer über alle Inputs. Kann einzeln passieren oder als Matrix-Multiplikation. (einzeln kann man sich es bisschen besser vorstellen). Unser Input (x1....xn) wird also wieder zu hidden States (h1...hn)

Das ist der Encoder.
Der Decoder ist eine GRU und bekommt als input die hidden States (c genannt)

Vorlesung 8

Was ist Self-attention? Was ist cross-attention? Unterschied?

Cross-attention hat ein X (source) und ein Y(target) für tasks like MT/seq2seq. Query Vektor wird aus Y erstellt, während k und v Input X repräsentieren. es wird ein scaled dot product zwischen q und k berechnet.

Self-attention besteht nur aus X. Kein Y. q,k,v alle aus Input token X. Hauptunterschied zwischen attention und self-attention: RNN fällt weg.

Was ist parallelized-attention?

Anstatt für jeden Input x_j einzeln das dot-Product zu berechnen, könnte man auch die Vektoren aufeinander stacken und die Matritzen multipizieren. (Gut für GPU ==> Faster). Dazu muss aber angenommen werden, dass neue Inputs nicht abhängig sind von alten outputs (Bei Bahdanau nicht möglich).
Berechnung quasi gleich. Nur mit Matritzen. Am Ende jede Zeile (q_j) Prob-Distr. über festgelegte Dimensionen. Dimensionen müssen übereinstimmen

$$\boldsymbol{O} = \mathcal{F}^{\text{attn}}(\boldsymbol{X}, \boldsymbol{Y}; \theta) = \text{softmax}\left(\frac{(\boldsymbol{Y}\boldsymbol{W}^{(q)})(\boldsymbol{X}\boldsymbol{W}^{(k)})^T}{\sqrt{d_k}}\right)(\boldsymbol{X}\boldsymbol{W}^{(v)})$$

Was ist Multilayer-attention?

Nutzen hier mehrere Attetion Layer !SEQUENTIELL!, die jeweils ihre eigene Parameter lernen. Im Transformer Model werden mehrere Blöcke von Transformern aufeinander angewandt.

Was ist Multihead-attention?

Mehrere (m) Attention Layers die !PARALLEL!. ihre eigenen Parameter auf V,K,Q anwenden. Die Ergebnisse (m Outputs) werden konkatiniert und mit einem neuen eigenen W in die richtige Dimension gebracht.

Was ist masked-attention? Warum/wann brauchen wir es?

Das Model kann sich zukünftige Inputs anschauen, was für Autoregressive LM cheaten bedeuten würde. Besser erklärt. Der output an Stelle j wird aufgrund aller v berechnet. ==> Müssen für o_j alle v_j' mit j'>j verdecken (alle zukünftigen)

wenn j'>j setzen wir einfach den alpha_j_j' auf 0. ==> v hat keinen Impact auf den output. (siehe Formel >) $\alpha_{j,j'}\boldsymbol{v}_{j'}$

Masked Self-attention wird im decoder genutzt, da wir hier nicht die Ergebnisse sehen wollen.

<u>Wie sieht parallel masked self attetion aus?</u> FOLIE 35 VL 8

Wenn wir keine Targets haben, brauchen wir einen Loop (Logisch, weil keine Matrix aufgebaut werden kann)

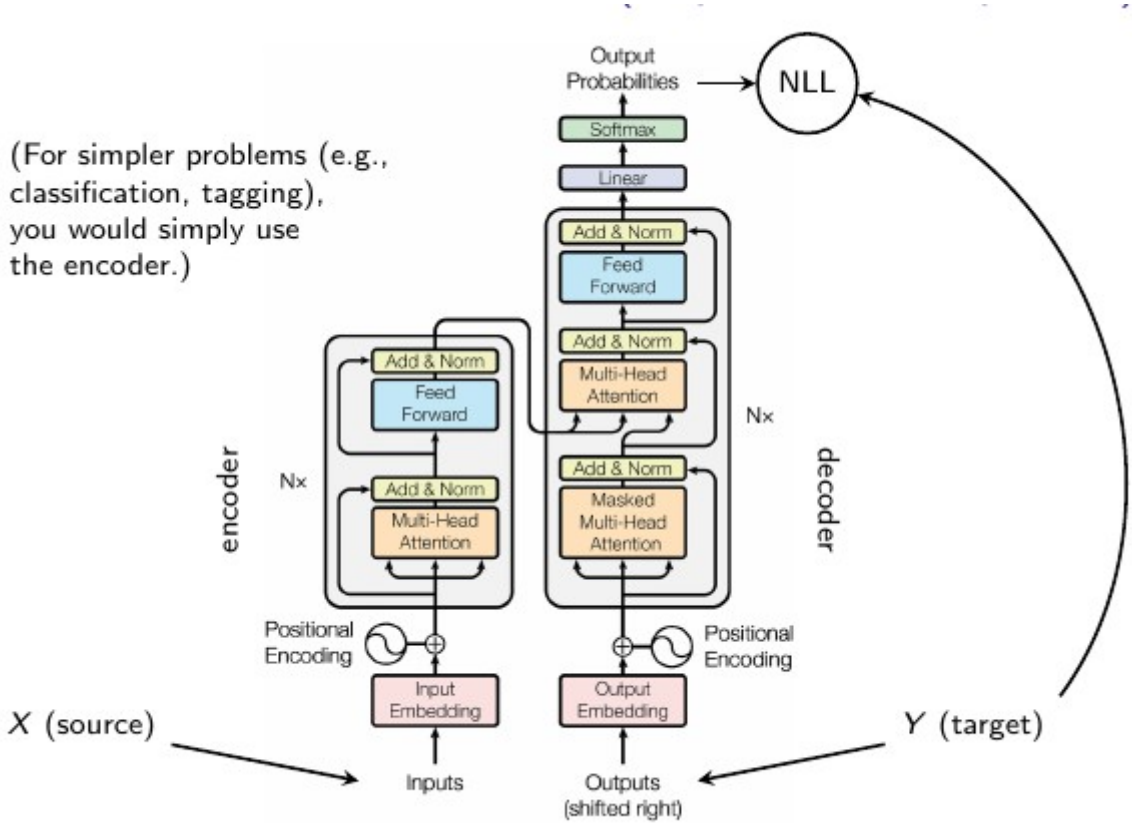
Erkläre Layer Normalization.

Normalisiert Vektoren, die aus einem Layer kommen. Hilft aktivations klein zu halten. Im transformer, wird jede Postion in O normalisiert.

Was sind Residual connections.

Man addiert X zum self-attention output um Informationen beizubehalten

Was gehört zur Transformer Architektur? Erkläre den Aufbau folgender Architektur:



Wozu dienen Position encodings und was sind sie?

Self attention ist es nicht möglich zw. "Mary loves John" und "john loves Mary" zu unterscheiden, da beide den gleiche Input bekommen und kein Layer/Funktion trägt zum Verständnis der Position im Satz bei.
Um das Problem zu lösen, fügen wir ein Positional Embeddiing zum WordEmbedding hinzu. Entweder trainable (begrenzte Satzgröße) oder sinusodial position Embeddings

Was sind Hyperparameter? Welche Hyperparameter kennen sie? Zählen Sie 5 auf. Related to training/ Related to the Model.

Wofür wird cross validation genutzt.

nicht genug Data. Teilen Trainingdaten in N gleich große Subsets. Trainieren ein Model mit n ausgenommen und evaluieren auf n. ==> Average über alle N

Erkläre Shifting Window algorithmus:

Nur für continuous Hyperparameter. Schauen uns immer ein Window (3) von möglichen Hyperparameter an. Sample Configurations aus dem Window. Setze die Mitte des Windows auf das Hyperparameter, das am besten performt hat ==> Repeat.

Erkläre Evolutionary algorithm:

Random Configurations - für n Iterationen beste Hyperparameter kombinieren und neue Konfigurationen daraus erstllen. Schlecht löschen wir.

Vorlesung 9

Was ist transfer learning? Wie funktioniert es?

die gelernten Informationen aus einer Aufgabe auf eine andere (verwandte) Aufgabe anzuwenden.

Was ist bytepair encoding?

dient zur kompression von daten. Schauen daten Byte Paar weise an und fassten häufigstes Paar mit unbenutztem Byte zusammen. ==> repeat until no compression possible (GPT3)

Was ist WordPiece?

wird in Asiatischen Sprachen benutzt, da sie mehr Chars haben und keine Leerzeichen + unterschiedliche Betonungen.

Sprache wird mit Unicode Chars dargestellt (sehr viele) + _ vor und oder nach Wort um Space darzustellen (==> Faktor * 4 für Vokabulary size)
--> Build LM ==> Merge Words die die Likelihodd maximieren.

In westlichen Sprachen 500 basic units.

Was ist SentencePiece?

BPE und WordPiece brauchen inputsequence. ==> SentencePiece braucht keine Leerzeichen Tokenisierung. Besteht aus bytepair encoding und unigramm LM

==> kein sprachspezifisches preprocessing!

Was ist das Problem mit Word2Vec?

Unbekannte Wörter haben kein Embedding (Fasttext). Unterschiedliche Bedeutungen des Wortes können nicht gespeichert werden (Bank)

Was sind die Basic Principals von Transfer Learning:

- generelle Aufgabe, viele ungelabelte Daten (self supervised)

Benutzen die gesammte Architektur und nutzen neues finales Layer für unsere Aufgabe, da Contextinformation nicht im Embedding sondern im Model gelernt wird.

Unterschied zw. inductive Transfer Learning (TL) und transductive TL:

- inductive. Haben Label nur in Target domain.
- transductive: haben Label nur in Source Domain

Was ist das besondere an ELMo bzw. seinen Embeddings?

2 Layer Bi LSTM ==> 4 context dependent Token representations + Token representation

Bei der Anwendung auf neue Task werden die 5 representations eingefrohren und 2 neue Parameter können gelernt werden. ==> haben zugriff auf 5 unterscheidliche "WordEmbeddings". Model lernt, welche Repräsentation wann wichtig ist.

Probleme mit ELMo?

embeddings wurden nicht taskspezifisch trainiert. LSTM < Tranformer.

Wie unterscheidet sich ULMFiT zu ELMo?

ist unidirectional. 3 LSTM Layers. Model kann auf eigenen Daten gefinetunt werden. (Elmo verändert WordEmbeddings nicht, sondern lernt nur 2 Parameter)

Wie unterscheidet sich GPT zu ELMo/ULMFit?

Tranformer architecture. 12 Layer decoder. Positional Embeddings.
Softmax Output. LM Objective. Kann mehrere Tasks lösen.

Was bedeutet Self-supervised Learning?

Haben unlabeled Data, aber bauen daraus beispiele/Aufgaben, die supervised sind. Zum Beispiel das nächste Wort in einem Satz vorherzusagen. (Language Modeling, Masked LM, Permutation LM ...

Was sind ein paar large-scale datasets, die sie benutzen können zum pretraining?

Wikipedia, Book Corpus, wikitext, commonCrawl..

Vorlesung 10:

Was ist Bert? Wie funktioniert Bert? Was ist der Unterschied zu ELMo ULMFiT usw.

neues self supervised objective MLM. Bidirectional Model. No recurrent architecture. Only self attention. Problem: Durch bidirectionality kann das Model das nächste Wort sehen.

Kann durch finetuning auch mehrere Task lösen

Byte-Pair Encoding. Token Emb, Segement Emb, Position Emb. 13 GB data. lange Trainingszeit. Loss = Loss(MLM) + Loss(NSP)

Elmo hat 2 unidirectional models genommen. GPT ist nicht bidirectional. BERT kombiniert self attention von GPT und Bidirectionality von ELMO.

Wie funktioniert Masked LM?

Only during pretraining. 15% der token werden durch ersetzt. 80% davon mit [MASK], 10% mit random token und 10% werden nicht verändert

Was passiert bei Next Sentence Prediction?

haben ein Seperator token zwischen zwei sätzen. 50% der Zeit random second Sentence, sonst tatsächlicher Satz. (vlg. negative sampling)

Was sind die Limitationen?

512 Tokengrenze. auch bei NSP. ==> Complexity von Transformern quadratisch.

Was hat ALBERT und RoBERTa anders gemacht?

- Roberta entfernt NSP, macht hyperparameter tuning, vergrößert Datenset, dynamic masking (= 10 fach duplizierter Korpus mit 10 unterschiedlichen MASKings.

- **AIBERT**: Verändert die Architektur. irgendwelche Context ab/unabhängigkeiten werden verändert (wie warum unklar). Attention parameter werden zwischen layer geteilt. Sentence Order Prediciton(SOP). NSP negative examples verändert nur die reihenfolge der Sätze. Ngram masking MLM.
==> kleineres Model und bessere Ergebnisse

Vorlesung 11:

Welche unterschiedlichen Trainingsziele gibt es?

Was sind Shortcomings von BERT und anderen.
Dadurch, dass das Training [MASK] benutzt und MASK nicht beim finetuning auftaucht, entsteht eine "Diskrepanz". Die vorhergesagten Token werden unabhängig vom Satz betrachtet. dh. eine gemeinsame WK fehlt.

Die maximale Inputlänge von 512 ist kurz. Laufzeit von self-attention ist quadratisch.

Autoregression: Likelihood über Produkt der bed. Wken.

AutoEncoding: Masked Tokens von corr. Seq. wieder herstellen.

Was waren die Änderungen an XLNET:
Benutzt relative segment/Position Encodings. Keine Independence Annahme
Permutation LM
==> Zwei Attention Mechanisms
- Query-stream: Hat Zugriff auf Context durch den Contentstream,aber keinen Zugriff auf current Position ==> Query Embedding
- Contenstream: Q,V,K von Contentstream ==> Content Embedding

Was waren die Änderungen an T5:

Encoder-Decoder Architektur (best for text2text). Text2text tasks. 11Bil. Parameters - Larger models trained for fewer steps better than smaller models on more data

Was waren die Änderungen an ELECTRA:

anderes Pretraining objective: Generator (MLM) + Discriminator(Replaced Token detection) ==> Lernt von allen Token

Welche anderen Architekturen gibt es. Wie wurden die Shortcomings gelöst?

Was ist model Distillation? Warum wollen wir kleinere Modelle?

geht darum ein Model kleiner zu machen. einfacheres deployment und bessere Generalisierung.

DistillBert

Viel kleiner (hälfte der layer, hälfte der Größe) gleichzeitig 95% Performance. Neuer Distillation Loss. kein NSP, dynamic masking + large batches

Problem der Quadratischen Laufzeit von Self-attention:

Weil alle Token miteinander verbunden werden. (+ Länge egal - Schlecht für lange Sätze) ==> Wollen ein tranformer Model, das das Problem löst. Gibt mehrere Modelle/Transformer, die das Problem lösen wollen und gleichzeitig die Qualität beibehalten. **DeBERTa Disentangled Attention, Linear-Transformer**

Vorlesung 12:

Probleme mit Bert/RoBERTa etc:

Löst nur eine Task (wollen alle(viele) Tasks lösen). Mensch braucht nicht Billionen von Beispielen um eine Aufgabe zu lernen (==> Few-Shot learning) . Anwendung in Realität kann ganz anders sein als auf dem Testset

Was ist GPT?

Language Model. Predicts next word (left to right). transformer architecture. single model to solve all tasks. kann Aufgaben beschreibungen nutzen. Few Shot learning

was ist few-shot/one-shot/zero shot learning?

Modell bekommt few/one/zero Beispiele + Aufgabenstellung. OHNE GRADIENTEN UPDATE

Welche Tasks gibt es? Wie funktionieren sie?

Lambada: Fill the blank.

QA: Question answering (ohne Text)

Winograd: Gegeben zwei Contexte und der Vervollständigung. Welcher Context ist der richtige?

ARC: gegeben eine Frage, welche von 4 Antworten ist die richtige (oft verständnis über die Welt von nöten)

RACE: Gegeben einen Artikel, beantworte Single-Choice Frage (GPT Kann nicht zurück in den Text gehen, nachdem es das Beispiel gelesen hat)

SuperGLUE: Samlung mehrerer Tests

- BoolQ: Given Text, answer yes or no to question
- WiC: Word in Context. Task, die sich um ambiguität bzw verwendung von einem Wort in unterschiedlichen Kontexten dreht.

- COPA: Choice of Plausible Alternatives: Sentence completion mit mehreren Wörtern (Nicht generieren sondern auswählen.)

WSC: Winograd Schema Challenge: Pronoun reference resolution. Wer ist im vorherigen Satz mit "er" gemeint?

- ReCoRD: Reading Comprehension with Commonsense Reasoning Dataset. article + mehrere Zusammenfassungen. Wähle die richtige Zusammenfassung aus.

ANLI: Given text and question decide True False Neither

SAT Analogies: Choose correct completion of context

==========

GPT Kann schlechte Grammatik korrigieren. Menschen können nicht unterschieden, ob GPT3 oder Mensch einen Artikel geschrieben hat. Aber die Details stimmen nicht (Jahreszahlen, Fakten usw) GPT3 kann Mensch und GPT3 unterscheiden.

Limitations von GPT3:

Text Generation (wiederholungen, wiedersprüche)
Common Sense (es fehlt wissen über die phyische Welt)
Comparison Tasks, Mensch ist besser und hat weniger Text gelesen als Maschine,
Zu groß um es in der Praxis zu nutzen. GPT3 lernt nur während Pretraining. Tasks trainieren es nicht. d.h. Gleiche Frage am nächsten Tag bringt keine Änderung.
Teures Training

Welche Probleme sprechen Marcus & Davis an?

Model hat kein biologisches, physikalisches, psychologisches, soziales Verständnis. Kann Objekte/Menschen nicht verfolgen. (Räumliches Vorstellungsvermögen)

Ethnische Probleme:
Model weiß nichts, dass seine Aussagen Folgen haben können (Suicide Prevention), Diskriminierend gegen bestimmte Menschen aufgrund von Bias in Daten. Kann für Schlechte Absichten eingesetzt werden, manche Jobs würden wegfallen.

GPT3 hat gender/race/religion bias