Documentatie Lucrare Practica

Implementare modele deep learning pentru traducere engleza-franceza

Tema lucrarii presupune implentarea si aplicarea practica a modelului transformer, acum o alegere arhipopulara pentru orice task ce necesita procesarea limbajului natural sub forma textuala si nu numai, dupa arhitectura originala propusa de Vaswani et al (2017) . Modelul a fost antrenat pe un set de date in format csv, alcatuit din 100.000+ perechi de cuvinte in engleza-franceza, disponibil open source. In final a fost inclus intr-o aplicatie web dezvoltata in python flask, unde poate fi folosit pentru traducere.

Modelul Transformer:

A apărut într-un context în care modelele tradiționale de traducere seq2seq bazate pe retele recurente se confruntau cu limitări semnificative în gestionarea dependențelor pe termen lung și în paralelizarea procesului de antrenament. Arhitectura propusă de Vaswani et al. (2017) a revoluționat aceste aspecte prin introducerea mecanismului de self-attention, care permite modelului să aloce dinamic importanță diferitelor părți ale secvenței de intrare în timp ce procesează fiecare element.

Această inovație a deschis noi orizonturi în domeniul traducerii automate și al altor taskuri de procesare a limbajului natural. În loc să proceseze secvențial datele de intrare, transformerele folosesc mecanismul de atenție pentru a evalua toate elementele dintr-o secvență simultan, rezultând astfel o creștere semnificativă a vitezei de antrenament și o mai bună capacitate de a captura relațiile pe termen lung în datele textuale.

De la introducerea sa, modelul Transformer a evoluat rapid și a fost adoptat în diverse aplicații NLP, de la traducerea automată la sumarizarea textelor, răspunsul la întrebări, și chiar în generarea de text. Un exemplu notabil al acestei evoluții este dezvoltarea modelelor de limbaj mari (large language models), cum ar fi GPT (Generative Pre-trained Transformer) și BERT (Bidirectional Encoder Representations from Transformers), care au demonstrat performanțe de top în multiple benchmark-uri NLP.

Arhitectura:

Această arhitectură este compusă din două părți principale: encoderul și decoderul.

Encoderul:

- **Stratul de încorporare (Embedding Layer)**: Primește ca intrare secvența de token-uri și o transformă în reprezentări vectoriale densificate, cunoscută și sub numele de încorporări de cuvinte (word embeddings).
- **Straturile de encoder (Encoder Layers)**: Acestea sunt multiple blocuri de encoder care operează asupra secvenței de încorporări. Fiecare bloc de encoder conține mai multe sub-straturi, cum ar fi:
    - **Multi-Head Self-Attention Layer**: Oferă encoderului capacitatea de a analiza relațiile dintre cuvinte în cadrul aceleiași propoziții.

- **Stratul de Feed-Forward (Feed-Forward Layer)**: Aplică o transformare liniară urmată de o funcție de activare neliniară, oferind encoderului capacitatea de a înțelege relațiile dintre cuvinte.
- **Straturile de normalizare (Normalization Layers)**: Acestea normalizează datele de ieşire ale fiecărui bloc de encoder, contribuind la stabilizarea şi accelerarea antrenamentului.

Decoderul:

- **Stratul de încorporare (Embedding Layer)**: Similar cu encoderul, transformă secvența de token-uri din limba țintă în încorporări vectoriale dense.
- **Straturile de decoder (Decoder Layers)**: Acestea sunt multiple blocuri de decoder care operează asupra secvenței de încorporări a limbii țintă. Fiecare bloc de decoder conține mai multe sub-straturi, inclusiv:
  - **Multi-Head Self-Attention Layer**: Ajută decoderul să se concentreze pe diferite părți ale secvenței de ieşire în timpul generării.
  - **Multi-Head Attention Layer**: Permite decoderului să atenționeze la encoderul de intrare pentru a încorpora informații contextuale în procesul de generare a traducerii.
  - **Stratul de Feed-Forward (Feed-Forward Layer)**: Similar cu encoderul, aplică o transformare liniară şi o funcție de activare neliniară pentru a înțelege relațiile dintre cuvinte în secvența de ieşire.
  - **Straturile de normalizare (Normalization Layers)**: Normalizează datele de ieşire ale fiecărui bloc de decoder.

Aceste componente lucrează împreună pentru a transforma o secvență de intrare într-o secvență de ieşire, oferind o abordare puternică şi flexibilă pentru sarcini de procesare a limbajului natural. Implementarea este facuta cu ajutorul tensorflow.

Aplicatia web :

Această implementare utilizează Flask, un cadru web uşor pentru Python, pentru a îngloba modelul de traducere automată într-o aplicație web accesibilă utilizatorilor.

Aplicația este simplă şi intuitivă: utilizatorii pot introduce textul în limba engleză în câmpul de text şi apoi pot apăsa butonul "Translate" pentru a obține traducerea în limba franceză. Apoi, traducerea este afişată în acelaşi formular.

Codul HTML al şablonului este minimalist şi oferă o interfață prietenoasă utilizatorilor. Interfața conține doar un câmp de text pentru introducerea textului de tradus şi un buton pentru a iniția procesul de traducere.

Concluzize:

Lucrarea de practica arata cum inovatiile teoretice din domeniul inteligentei artificiale pot fi implementate pentru a rezolva o porblema atat de cruciala in lumea interconectata in care traim astazi. Modelul transformer, cu capabilitatile lui nemaivazute de scalare este piatra de temelie pentru evolutia puternica a domeniului din ultimii ani.

Codul sursa este disponibil pe github https://github.com/simondarius/Flask-Translator .