Optymalną architekturę modelu wybraliśmy porównując wartości BCELoss (Binary Cross-Entropy Loss) dla zbioru walidacyjnego. Najniższa wartość (uzyskana w dowolnym epochu) wskazywała na najlepszy model. Zmian dokonywaliśmy w parametrach [TUTAJ PARAMETRY, KTÓRE ZMIENIAŁEŚ PRZY SZUKANIU HIPERPARAMETRÓW wraz z uzasadnieniem wyboru hiperparametrów takich jak liczba i typu warstw czy rozmiary osadzeń i jaki (rozmiarowo) encoding jest najlepszy dla klasyfikatora].

Klasyfikator trenowany był wg algorytmu:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznieZapobiegliśmy przeuczeniu poprzez zachowanie jedynie modelu o najniższym BCELoss dla zbioru walidacyjnego i zatrzymując trenowanie, jeśli 10 razy z rzędu po uczeniu model nie będzie lepszy od najlepszego modelu.

Obraz zawierający tekst, zrzut ekranu, Wykres, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Wykres, linia, diagram

Opis wygenerowany automatycznieRozłożenie danych i granica decyzyjna przy kodowaniu w jednym wymiarze dla najlepszej uzyskanej dokładności na zbiorze walidacyjnym (lewo) oraz wykres wartości MSELoss (prawo):

Jak można łatwo zauważyć wartości znacznie nachodzą na siebie; nie jest to dobra metoda kodowania, ponieważ trudno jest oddzielić od siebie klasy.Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, Wykres, diagram, linia

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Wykres, diagram

Opis wygenerowany automatycznieRozłożenie danych i granica decyzyjna przy kodowaniu w 2 wymiarach dla najlepszej uzyskanej dokładności na zbiorze walidacyjnym (lewo) oraz wykres wartości MSELoss (prawo):

Wyraźnie widać, że podział jest znacznie lepszy od 1-wymiarowego, lecz punkty nadal nie są wyraźnie rozdzielone.

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Najlepszy wynik dokładności uzyskaliśmy jednak dla embeddingu 128 wymiarowego:  
Obraz zawierający Wykres, tekst, linia, diagram

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

[Tu kod najlepszego modelu (nie napisałeś w docx który to)]

Optymalną architekturę modelu wybraliśmy porównując wartości MSELoss (Mean Squared Error Loss) dla zbioru walidacyjnego. Najniższa wartość (uzyskana w dowolnym epochu) wskazywała na najlepszy model. Zmian dokonywaliśmy w parametrach [TUTAJ PARAMETRY, KTÓRE ZMIENIAŁEŚ PRZY SZUKANIU HIPERPARAMETRÓW wraz z uzasadnieniem wyboru hiperparametrów takich jak liczba i typ warstw czy rozmiary osadzeń i jaki (rozmiarowo) encoding jest najlepszy dla modelu regresji].

Model regresji trenowany był zgodnie z poniższym algorytmem:  
Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Zapobiegliśmy przeuczeniu poprzez zachowanie jedynie modelu o najniższym MSELoss dla zbioru walidacyjnego i zatrzymując trenowanie, jeśli 10 razy z rzędu po uczeniu model nie będzie lepszy od najlepszego modelu.

Najniższą wartość MSELoss w przypadku kodowania w 1 wymiarze uzyskaliśmy dla modelu liniowego (TransformerLinearRegressionModel):

Obraz zawierający tekst, Wykres, linia, zrzut ekranu

Opis wygenerowany automatycznieObraz zawierający tekst, Wykres, zrzut ekranu, linia

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieJak wyraźnie widać uzyskaliśmy słaby wynik.  
Jeszcze gorsze wyniki uzyskaliśmy dla modelów nieliniowych (Transformer NonlinearRegressionModel) przedstawionego poniżej:

Obraz zawierający tekst, zrzut ekranu, diagram, Wykres

Opis wygenerowany automatycznieObraz zawierający tekst, Wykres, linia, diagram

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznieDla kodowania w 2 wymiarach osiągnięto najlepsze wyniki przy użyciu TransformerNonlinearRegressionModel:

Jednak najlepsze wyniki osiągnięto dla kodowania w 128 wymiarach z użyciem TransformerLinearRegressionModel: Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznieObraz zawierający tekst, Wykres, linia, diagram

Opis wygenerowany automatycznieDla modelu linowego rozkład w 2 wymiarach wyglądał następująco:

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie