Na początek modyfikacja modelu..

#Model

class NGramModel(nn.Module):

    def \_\_init\_\_(self, vocab\_size, embedding\_dim, context\_size):

        super().\_\_init\_\_()

        self.embeddings = nn.Embedding(vocab\_size, embedding\_dim)

        self.linear1 = nn.Linear(context\_size \* embedding\_dim, HD)

        self.linear2 = nn.Linear(HD, vocab\_size)

    def forward(self, inputs):

        embeds = self.embeddings(inputs).view((1, -1))

        print("pred embending: ", embeds) 🡨|

        out = F.relu(self.linear1(embeds))   |

        print("pred relu: ", out) 🡨| Dodane printy żeby

        out = self.linear2(out) |---zobaczyć co się dzieje

        print("pred linear2: ", out) 🡨| na poszczególnych

        log\_probs = F.log\_softmax(out, dim=1)  | etapach przewidywania

        print("pred log\_probs: ", log\_probs) 🡨|

        return log\_probs

Później kilka zestawień:

# dla słowa 'the' : 53

# dla słowa 'way' : 33

# dla słowa 'to': 5

# embeddingi

print(list(model.parameters())[0][53])

tensor([ 0.2481,  0.3692, -0.7330,  0.4940, -0.8680,  0.7258, -0.5098, -1.2559, 0.8786, -0.4926], grad\_fn=<SelectBackward>)

print(list(model.parameters())[0][33])

tensor([ 0.0186,  0.6130,  1.4379, -0.0066, -0.7198,  0.4909, -0.9187, -1.0668, 0.4317,  1.1668], grad\_fn=<SelectBackward>)

print(list(model.parameters())[0][5])

tensor([ 1.0905,  0.4925,  0.2505,  0.5815, -1.2171,  1.0470, -0.0308, -0.1842, -0.1207,  0.9539], grad\_fn=<SelectBackward>)

Wywołanie “ pred = model(context\_idxs) “ wyrzuciło kolejne tensory przeliczane w modelu.

Pierwszy z nich uzyskany przez dodatnie emendingów the(53) i way(33):

pred embending: tensor([[ 0.2481,  0.3692, -0.7330,  0.4940, -0.8680,

0.7258, -0.5098, -1.2559, 0.8786, -0.4926,

0.0186,  0.6130,  1.4379, -0.0066, -0.7198,

0.4909, -0.9187, -1.0668,  0.4317,  1.1668]])