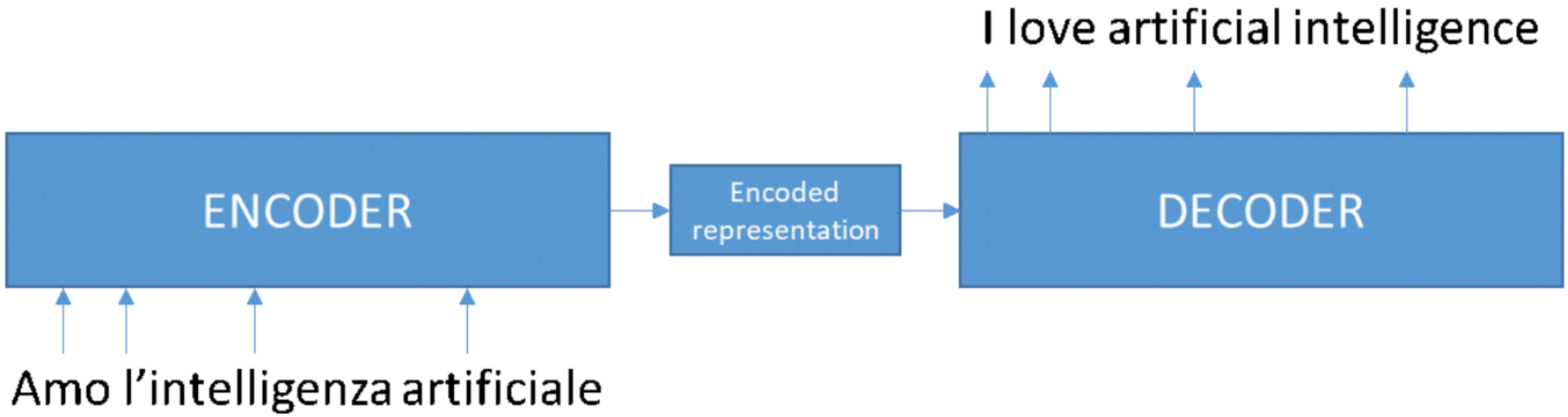
Sequence to Sequence Model using RNN

Sequence-to-sequence learning

- Sequence-to-sequence learning (Seq2Seq) is about training models to convert sequences from one domain (e.g. sentences in English) to sequences in another domain (e.g. the same sentences translated to French)

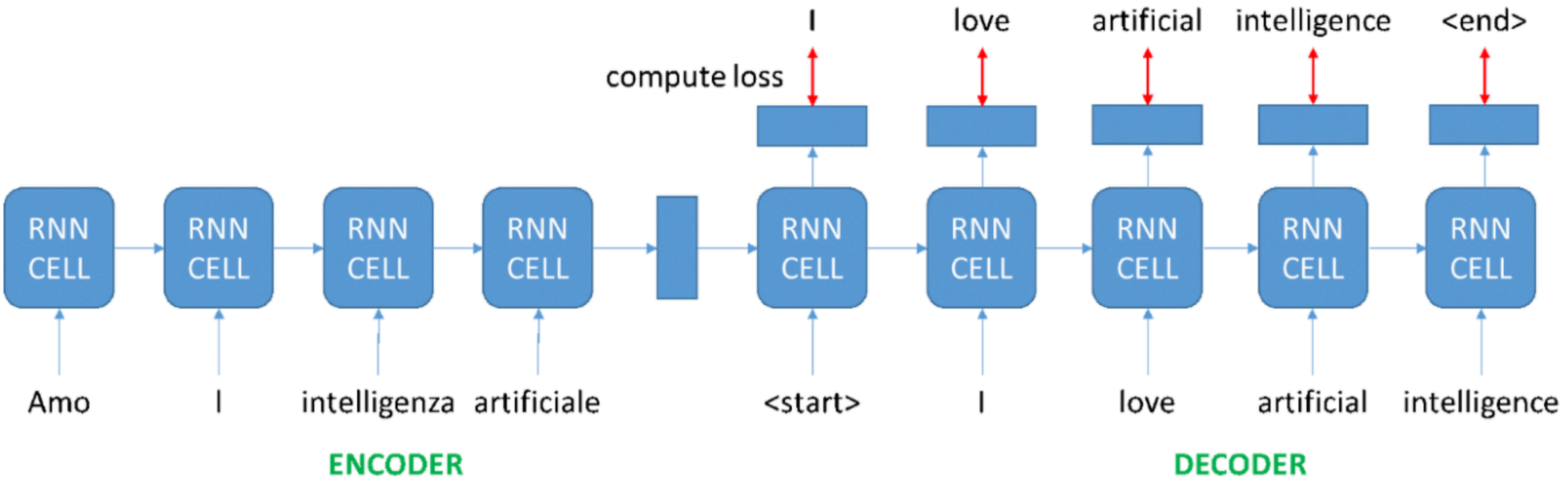
Encoder Decoder Network



symbolic scheme of an encoder-decoder architecture

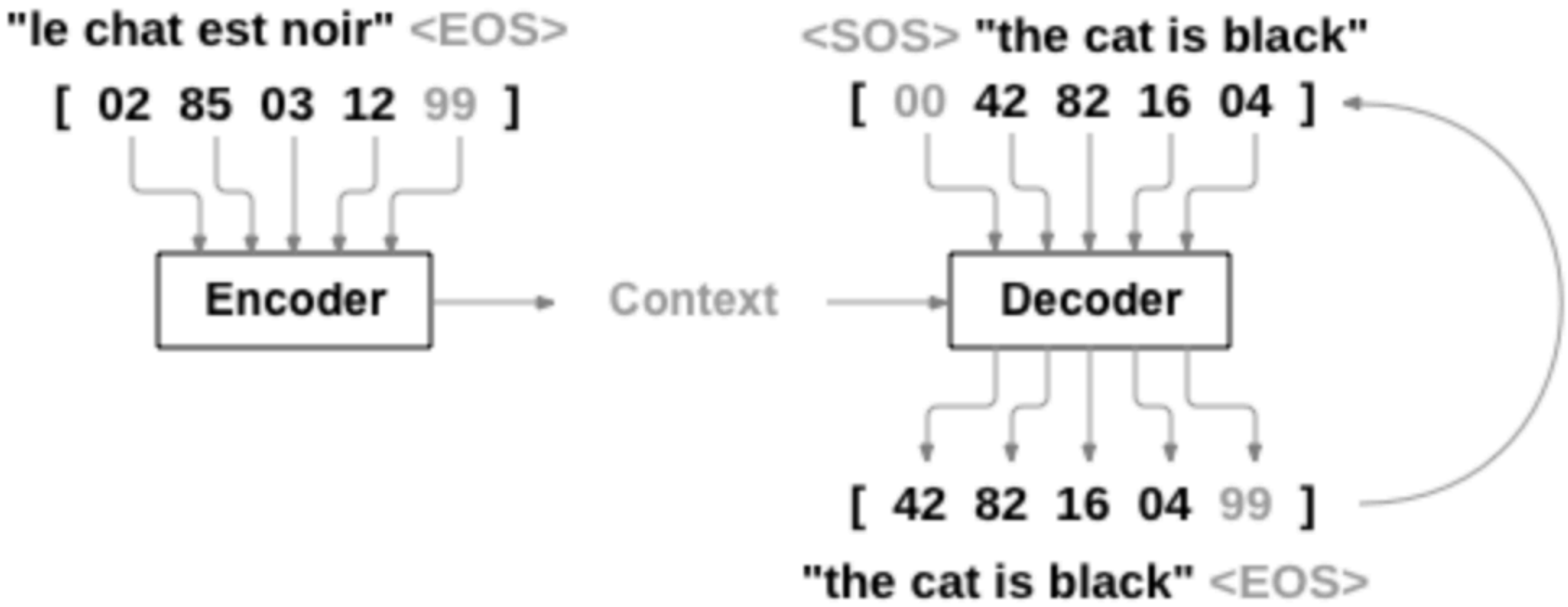
https://medium.com/@d.salvaggio/sequence-to-sequence-architectures-ad6ff4451f84

Encoder Decoder Network



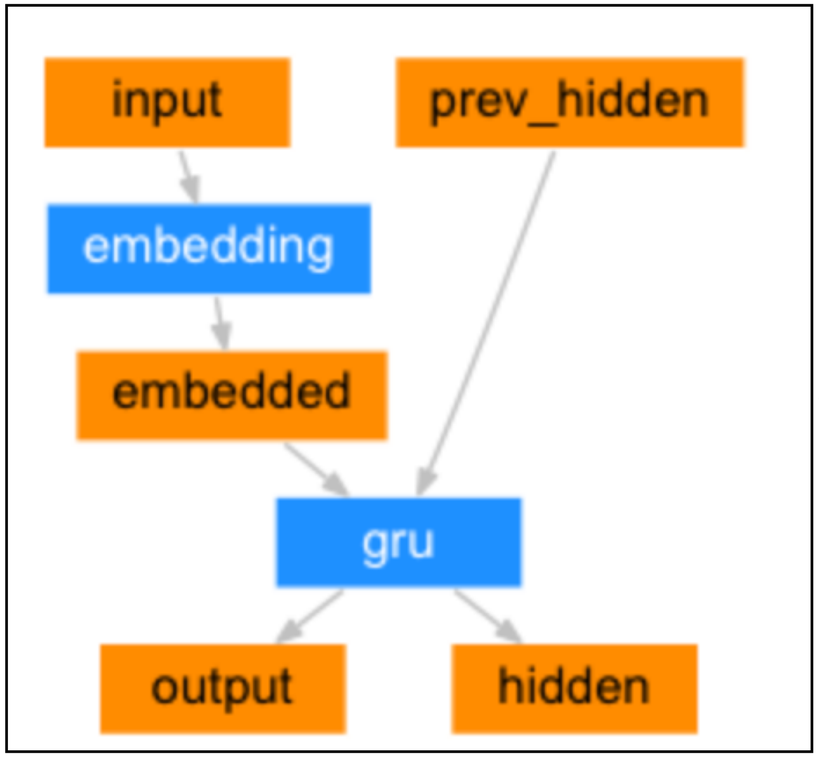
https://medium.com/@d.salvaggio/sequence-to-sequence-architectures-ad6ff4451f84

French to English Translation



https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

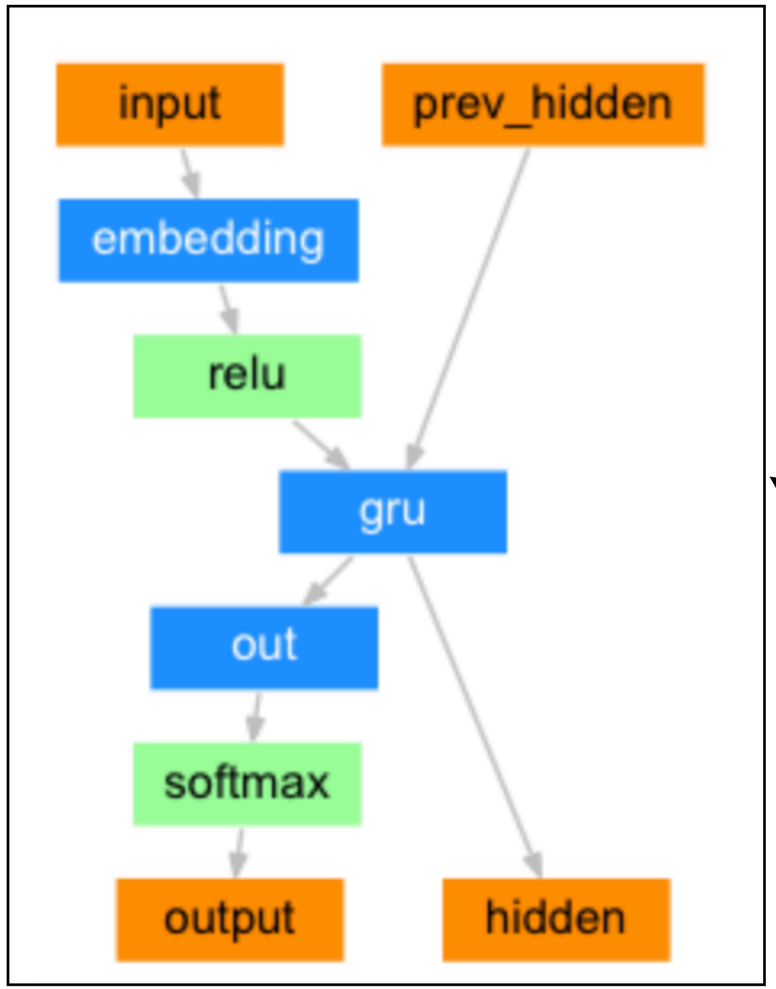
Encoder




```
class EncoderRNN(nn.Module):
    def __init__(self, input_size, hidden_size):
        super(EncoderRNN, self).__init__()
        self.hidden_size = hidden_size

        self.embedding = nn.Embedding(input_size,
hidden_size)
        self.gru = nn.GRU(hidden_size, hidden_size)

    def forward(self, input, hidden):
        embedded = self.embedding(input).view(1, 1, -1)
        output = embedded
        output, hidden = self.gru(output, hidden)
        return output, hidden

    def initHidden(self):
        return torch.zeros(1, 1, self.hidden_size,
device=device)
```

https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

Decoder




```
class DecoderRNN(nn.Module):
    def __init__(self, hidden_size, output_size):
        super(DecoderRNN, self).__init__()
        self.hidden_size = hidden_size

        self.embedding = nn.Embedding(output_size,
hidden_size)
        self.gru = nn.GRU(hidden_size, hidden_size)
        self.out = nn.Linear(hidden_size, output_size)
        self.softmax = nn.LogSoftmax(dim=1)

    def forward(self, input, hidden):
        output = self.embedding(input).view(1, 1, -1)
        output = F.relu(output)
        output, hidden = self.gru(output, hidden)
        output = self.softmax(self.out(output[0]))
        return output, hidden

    def initHidden(self):
        return torch.zeros(1, 1, self.hidden_size,
device=device)
```

https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html

Lab - Sequence to Sequence Model using RNN

Click this link to begin

https://colab.research.google.com/github/ravi-ilango/acm-dec-2020-nlp/blob/main/lab1/seq2seq_translation_tutorial.ipynb

Based on https://pytorch.org/tutorials/