**Dlaczego tradycyjne rekurencyjne sieci neuronowe (RNN) mogą mieć problemy z przechowywaniem informacji na dłuższe okresy, a jakie innowacje wprowadza architektura LSTM w celu rozwiązania tego problemu?**

Tradycyjne RNN mogą doświadczać problemu z zanikającymi gradientami, co prowadzi do utraty informacji na dłuższe okresy. Gdy sieć jest szkolona na długich sekwencjach, gradienty, które są przekazywane z końca sekwencji do początku, mogą się zanikać, co utrudnia efektywne uczenie się na odległych zależnościach czasowych. Architektura LSTM wprowadza dodatkowe bramki (forget gate, input gate, output gate), które regulują przepływ informacji, zapobiegając zanikaniu gradientów i umożliwiając przechowywanie istotnych informacji na dłuższe okresy.

Rozważmy prosty model rekurencyjny, który przetwarza sekwencję danych liczbowych. Niech będzie to jednokierunkowa rekurencyjna sieć neuronowa z jednym neuronem w warstwie ukrytej. Dane wejściowe to sekwencja liczb: [2, 4, 1, 3, 5]. Każda liczba jest przetwarzana w kolejnych krokach czasowych. Załóżmy, że wagi są równe 0.5 dla wag między warstwą wejściową a ukrytą oraz 0.8 dla wag między warstwą ukrytą a nią samą. Początkowy stan ukryty wynosi 1.

Oblicz stan ukryty po przetworzeniu całej sekwencji danych.

Oczywiście, przepraszam za wcześniejsze zamieszanie. Oto poprawione obliczenia dla sekwencji ATGCC:

Dane wejściowe (ATGCC): �1=1, �2=2, �3=4, �4=3, �5=4*x*1​=1,*x*2​=2,*x*3​=4,*x*4​=3,*x*5​=4

Parametry: �=[0.5,1], �=−2*W*=[0.5,1],*b*=−2

Początkowy stan ukryty: ℎ0=0*h*0​=0

Obliczenia (dla każdego kroku czasowego):

1. **Krok czasowy 1:**
   * �1=�⋅[ℎ0,�1]+�=[0.5,1]⋅[0,1]−2=(0.5×0+1×1)−2=−1*z*1​=*W*⋅[*h*0​,*x*1​]+*b*=[0.5,1]⋅[0,1]−2=(0.5×0+1×1)−2=−1
   * ℎ1=ReLU(�1)=ReLU(−1)=0*h*1​=ReLU(*z*1​)=ReLU(−1)=0
2. **Krok czasowy 2:**
   * �2=�⋅[ℎ1,�2]+�=[0.5,1]⋅[0,2]−2=(0.5×0+1×2)−2=0*z*2​=*W*⋅[*h*1​,*x*2​]+*b*=[0.5,1]⋅[0,2]−2=(0.5×0+1×2)−2=0
   * ℎ2=ReLU(�2)=ReLU(0)=0*h*2​=ReLU(*z*2​)=ReLU(0)=0
3. **Krok czasowy 3:**
   * �3=�⋅[ℎ2,�3]+�=[0.5,1]⋅[0,4]−2=(0.5×0+1×4)−2=2*z*3​=*W*⋅[*h*2​,*x*3​]+*b*=[0.5,1]⋅[0,4]−2=(0.5×0+1×4)−2=2
   * ℎ3=ReLU(�3)=ReLU(2)=2*h*3​=ReLU(*z*3​)=ReLU(2)=2
4. **Krok czasowy 4:**
   * �4=�⋅[ℎ3,�4]+�=[0.5,1]⋅[2,3]−2=(0.5×2+1×3)−2=2*z*4​=*W*⋅[*h*3​,*x*4​]+*b*=[0.5,1]⋅[2,3]−2=(0.5×2+1×3)−2=2
   * ℎ4=ReLU(�4)=ReLU(2)=2*h*4​=ReLU(*z*4​)=ReLU(2)=2
5. **Krok czasowy 5:**
   * �5=�⋅[ℎ4,�5]+�=[0.5,1]⋅[2,4]−2=(0.5×2+1×4)−2=2*z*5​=*W*⋅[*h*4​,*x*5​]+*b*=[0.5,1]⋅[2,4]−2=(0.5×2+1×4)−2=2
   * ℎ5=ReLU(�5)=ReLU(2)=2*h*5​=ReLU(*z*5​)=ReLU(2)=2

Ostateczny stan ukryty po przetworzeniu sekwencji ATGCC wynosi ℎ5=2*h*5​=2.