Parameter Values and Fatigue Mechanisms for FLIF Neurons

Christian Huyck
Middlesex University,
London, UK
c.huyck@mdx.ac.uk

and

Artemis Parvizi
Middlesex University,
London, UK

Abstract. A typical human brain consists of roughly 100 billion neurons, and one key aim of Biological Cybernetics is to simulate neural systems. A good model of a neuron accurately represents the behaviour of biological neurons, typically the spiking behaviour. For cybernetic systems that aim to function in real time with thousands, millions, or even billions of simulated neurons, it is also important that the model is computationally efficient. One Fatiguing Leaky Integrate and Fire neuron is a model that has four free parameters per neuron. This model has been used in cybernetic agents, but there have been few links to actual biological behaviour. A model of a rat neocortical neuron is developed with four specific parameter settings. This model is tuned to a particular input regime. When compared to a biological neuron it gets 90% of spikes roughly correct. Further modifications of the fatigue model enables the FLIF neuron to account for spontaneous neural firing, a known neural property, that is not present in the data. These modifications provide other FLIF models with a similar fit to the biological data. The best of these models correctly predicts over 94% of the spikes.

Keywords: Neural Model, Point Neuron, Integrate and Fire Neuron, Biological Fit, Accommodation, Spontaneous Activation.

1 Introduction

One method for exploring Biological Cybernetics is to build simulated neural systems. At one extreme, this approach has led to a project hoping to simulate the entire human brain [26]. Another framework is to build increasingly sophisticated systems from simulated neurons [12, 17]; this framework builds cognitive models and agents that function in an environment.

One of the key questions for Biological Cybernetics is what neural model or models to use. There are a large number of neural models (see section 5), and many researchers think that non-neural connectionist models are also a good basis (e.g. Multi Layer Perceptrons [29] or Self Organising Maps [23]).

One framework uses a simple point model of a neuron, the Fatiguing Leaky Integrate and Fire (FLIF) neuron (see section 2). There has been significant progress within this framework including simulated games agents [14] with vision, planning, action and language, machine learning [13], and cognitive models of natural language parsing [15] and task selection [4]. These have all used the same FLIF neural model. An individual neuron has several internal parameters including , firing threshold, leakage rate, and fatigue rates.

One key behaviour of neurons and neuron models is spiking. Neurons receive inputs from other neurons, and emit spikes. These spikes can be induced in biological neurons both *in vivo* and *in vitro*. For example, a neuron can be directly stimulated by electrical current, its internal electrical state can be measured, and spikes can be inferred (see section 3). Consequently, one way of calibrating a neural model, is to compare its spiking behaviour to biological data. In this paper, biological data is used to derive parameter settings for the FLIF neural model, and gauge its accuracy as a model. Then the fatigue model is extended to provide better fit and support for spontaneous neural firing.

2 Fatiguing Leaky Integrate and Fire Neurons

The FLIF model is an extension of the leaky integrate and fire (LIF) model, which itself is an extension of the Integrate and Fire model. A model that is similar to the FLIF model described below has been shown to account for inter-spike intervals under various input conditions better than the standard LIF model[7].

One variant of the Integrate and Fire (IF) model is the McCulloch Pitts neuron [27], which has a long standing history and is quite simple. Roughly, neurons are connected by uni-directional synapses. A neuron integrates activity from the synapses connected to it, and if the activity surpasses a threshold, the neuron fires, sending activity to the neurons to which it connects. There are two possibilities regarding retaining activity between cycles. The McCulloch Pitts neuron merely throws it away; this prevents low amounts of input causing the neuron to fire. In a second and earlier model [1], all of the activity is retained; this allows small amounts of input collected over a long time to cause the neuron to fire.

In the LIF model, if a neuron does not fire, it retains a portion (but not all) of its activity making it easier to fire later [25], with the lost activity leaking away. Typically, both IF and LIF neurons lose all activity when they fire.

The LIF model is extended to FLIF by the addition of fatigue. When a neuron fires, it fatigues and becomes more difficult to fire.

One of the major components of the model is the firing threshold, θ. A neuron i fires if its activation A_i minus its fatigue F_i is above the threshold.

$$A_i - F_i >= \theta \tag{1}$$

If the neuron fires, it loses all its activation. If sufficient activation is provided from neurons sending spikes to it, it may fire in the next time step.

If a neuron does not fire, some of its activation leaks away. This leak, or decay, is represented by a constant D where $D > 1$. Ignoring external input and assuming i did not fire at $t-1$, activation of neuron i at time t is

$$A_i^t = A_i^{t-1}/D \tag{2}$$

When neuron i fires, it sends activation (or inhibition) along its synapses to other neurons according to the strength of each synapse, so neuron j receives activation according to synaptic strength w_{ij}. The neuron is an integrator, so it accumulates activity from the synapses connected to it. So, given P_j^t, the prior activation of neuron j at time t, either 0 or Eq. 2, the activation at time $t+1$ is

$$A_j^{t+1} = P_j^t + \sum_{i \in V_i} w_{ij} \tag{3}$$

where V_i is the set of all neurons that fired at time t.

These equations describe a LIF model [25], and fatigue is used to extend the model. Fatigue uses two constants; it is incremented by F_c in a cycle when the neuron fires (Eq. 4), and is decremented by F_r in a cycle when the neuron does not fire (Eq. 5). $F_i >= 0$, so that firing always requires at least θ retained activation. Accumulated fatigue makes it more difficult for neurons to fire when they have been firing at a high rate.

$$F_i^{t+1} = F_i^t + F_c \tag{4}$$

$$F_i^{t+1} = F_i^t - F_r$$
$$(F_i^{t+1} < 0) \rightarrow F_i^{t+1} = 0 \tag{5}$$

The model has a loose link with time in biological neurons. The model does not incorporate conductance delays or refractory periods, and these behaviours all happen in under 10 ms., so each given cycle can be considered to be roughly 10 ms. Consequently, each neuron emits at most one spike per 10 ms. of simulated time, and the timing precision is at most 10 ms. This is a shortcoming of the model, but enables efficient simulation of hundreds of thousands of neurons in real time on a standard PC. It is consistent with the neural data modelled in this paper (see section 3), as the neuron being modelled never spikes more than once in a 10 ms. interval.

3 Neural Data

The neural data was used for a neural modelling competition [5], and the data was from Challenge A of that competition. More details can be found there along with the data. Note that this is one of a growing number of such benchmarks [18] that can facilitate comparisons between neural models.

The neuron was extracted from the primary somatosensory neocortex of a rat. So, the data that was collected was *in vitro*.

A probe was placed into the neuron. Current was injected directly to the neuron and cellular voltage was measured at .1 ms. intervals.

Input varied over 60 seconds, but only the first 38 seconds were available with the remaining 22 seconds used as the test for the competition. There was an initial input phase, followed by two seconds of 300mV input, then two seconds with no input, then two second of 600mV input, then two seconds with no input, then two seconds of 900mV input, then two seconds of no input, followed by 42.5 seconds of stochastic input, with 20.5 of that available as data. The same input regime was applied 13 times to the neuron.

This input is shown in Figure 1. Each point shows the input level to the model at that time. It has been converted from the original data in .1ms intervals to 10ms by averaging as described in section 4.

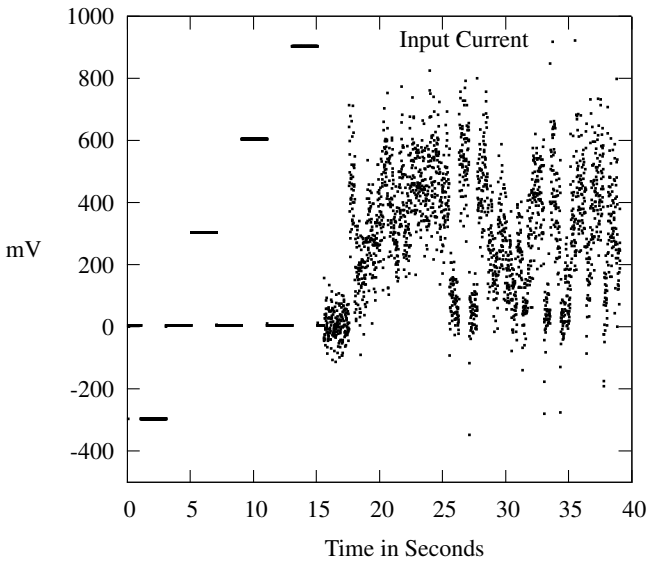

Figure 1. Input to the model converted from the current for the biological test

Also, the cellular voltage measurements were converted to spikes. With no input, the voltage hovered around -65mV. Under input this gradually increased. When it neared -40mV it then rapidly increased to positive values around 40mV. It then rapidly returned to a negative value, and then more gradually to a value around -55mV while under input. The spike was calculated from the first point the voltage crossed from negative to positive. It then was reset when it crossed back to negative. In the data, the shortest inter-spike interval was 13.5 ms. These spikes are shown in Figures 2 and 4.

4 Simulations and Parameter Settings

The initial task was to discover appropriate parameter settings for the FLIF model for input, the threshold θ, leak D, fatigue F_c, and fatigue recovery F_r. The parameters interact in relatively simple ways, so initially the goal was to find one parameter set, where the set led to model behaviour that was a relatively close fit to the biological data. All code and data can be found at http://www.cwa.mdx.ac.uk/chris/hebb/FLIF/FLIF.html

Clamped Input

The first question was how to scale the input. As the first input of interest was 300mV at a rate of .1 ms., the input was averaged over

the full time of the step (10 ms. or 100 pieces of data), and converted to units in volts. For example, input at time steps 50,000 to 50,100 (5 to 5.1 seconds) were all 300mV, and this is converted to .3 for input.

Next some analysis of the first series of inputs was used to set initial parameters. This series of inputs was two seconds of 300mV. Analysis of the biological data of one test of the neuron showed that there were 12 spikes at an average of 174.5 ms. apart. For the FLIF model, that was every 17 cycles. Using data derived from other studies [24], D was set at 1.25. These values were used to determine when the threshold was passed at 17 cycles, $\theta = 1.46$. These parameter settings led to data that fit the 300mV input. However, the parameters also needed to work for the other inputs.

Two other sets of input were of considered next; after rest periods of two seconds, there were two seconds of inputs at 600mV, then two seconds rest, and two seconds of 900mV. Some analysis of this data showed that the neuron spiked on average every 60.38 ms. for 600mV and every 40.51 ms. for 900mV, or every six and four FLIF cycles respectively. With $\theta = 1.46$ and $D = 1.25$ the model led to firing every three steps for 600mV and every two steps for 900mV. Reducing D, retaining more activation per step, moved things in the right direction for 600mV and 900mV.

Setting $D = 1.1$ led to values that worked reasonably well. Using the process described above for 300mV, θ was calculated at 2.6. This led to the desired behaviour with spikes every 17 cycles for 300mv, every 6 cycles for 600mV, and every four cycles for 900mV.

These parameters determine a LIF model. Perhaps fatigue was unnecessary. Some further analysis of the data showed that fatigue would improve fit. Figure 2 shows the latencies, time between spikes, of the neuron under the three input regimes. After initial rapid firing, all three spike latencies are relatively stable. However, there is a gradual increase at 900mV, increasing on average of .329 ms. between each spike after the third spike; that is, the spikes are coming more slowly, implying that at this elevated firing rate fatigue has an effect. At the lower firing rates, the slope is virtual flat, so the rate remains roughly constant.

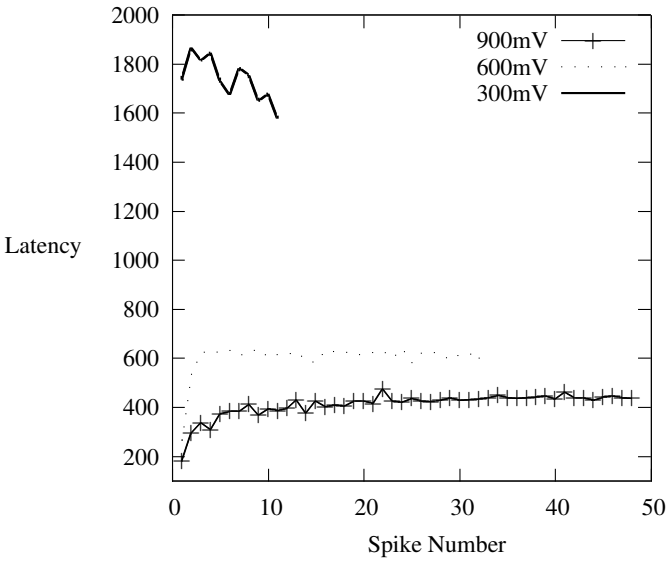

Figure 2. Latencies between biological spikes at different input values over two seconds

To calculate the fatigue F_c and fatigue recovery F_r values, this behaviour imposes some constraints. Rates at less than once per six cycles should not accumulate fatigue, and rates at once per four cycles should. So $F_c > 3F_r$, and $F_c < 5F_r$.

For the 900mV case, firing rates are initially just over every 35 ms., and they pass every 45 ms. around spike 35. Expanding the formulas with $\theta = 2.6$ and $D = 1.1$ shows a neuron having 3.138 units of activation at any fourth cycle. It needs 2.6 to fire, so it has 0.538 surplus activity. For accumulated fatigue to cause it pause for another cycle, it must be greater than .538. As this should not accumulate for 35 cycles, $F_c - 3F_r \sim (.538 \div 35)$. So, $F_c - 3F_r \sim 0.015$.

F_r was selected as 0.1, leaving F_c as 0.315. Running simulations on this showed that indeed spike rates at this the 35th cycle were every five cycles, but they returned to every four cycles thereafter. Reducing F_r to 0.01 left F_c at 0.045. The model latencies increased to five cycles after 35 spikes and continued to increase thereafter. The model predicts that after 200 spikes the latency would increase to six cycles, a testable hypothesis though not in the data.

This leaves all parameter values determined. Threshold is $\theta = 2.6$, leak is $D = 1.1$, fatigue is $F_c = 0.045$ and fatigue recovery is $F_r = 0.01$.

Stochastic Input

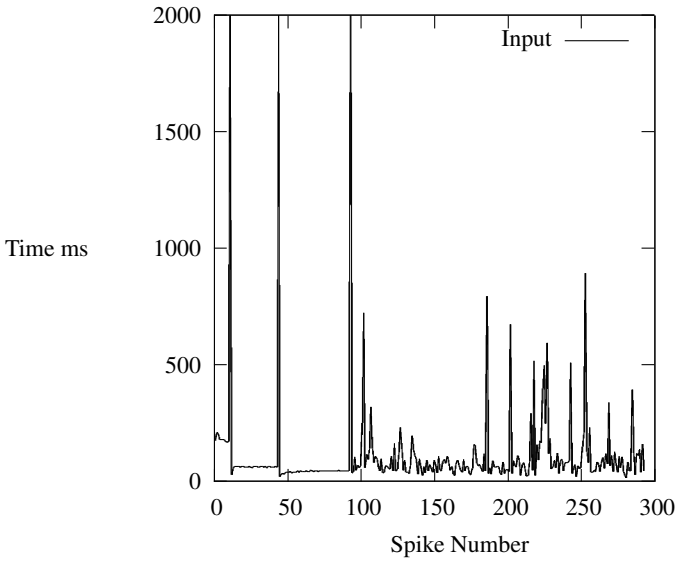

Figure 3. Interspike latency of the biological neuron

The first 17.5 seconds of input was clamped with either -300mV, 0mV, 300mV, 600mV or 900mV, and the stimuli persisting for two seconds, with intervals of no input in between. After this, the remaining 21.5 seconds of input went through a rapidly varying stochastic input. Figure 3 shows the response of the biological neuron to these inputs in the form of inter-spike intervals (another version can be seen in Figure 4 with model spikes). The first 11 spikes come around 170 ms. apart, followed by two seconds with no input, and thus no spikes. After spike 93 and the third two second delay, varying stochastic input began. Note that many periods during stochastic

input experience low input and thus high spike latency.

Using the same parameters, the model was run to account for this behaviour. For the biological neuron, input varied every .1 ms.; as the simulation steps accounted for 10 ms. of time, the inputs were binned into groups of 100 and averaged over that time.

During the clamped period, the model produced 92 spikes, and the biological data 94. The missing spikes were during the 900mV input.

During the stochastic period, the model produced 151 spikes and the biological data produced 200. The model spikes and real spikes were aligned, with the model spikes being placed adjacent to the nearest real spike. Analysis of these aligned spikes showed three categories of problems.

The first problem was evident even under clamped input. The first biological spikes after input resumed came earlier than the model predicted. This is evident in Figure 2 when the initial spikes are quite rapid, but the first spikes are even more rapid; for example at 300mV, the first spike occurs 36 ms. after input begins. This is also evident in the stochastic period, when model incorrectly does not produce spikes after relatively long periods of low or negative input are followed by moderate or high input.

The second problem was that periods of rapid biological spiking led to missing modelled spikes. This implied that the threshold was too high in relation to the other parameter settings.

The third problem was that spikes were missed over periods of low input. This implied that there was too much leakage in the parameter settings.

A further search of the parameter space ensued. Reducing the threshold and decay separately or together led to improved behaviour under stochastic input, but worse behaviour under clamped input. In particular, reducing decay requires increasing threshold, which had an adverse effect when there is high input. On the other hand, moderate increases of decay (e.g. $D \sim 1.2$) made it difficult to spike under low input (e.g. 300mV). Reducing threshold to allow this, made it spike too much under higher input.

Setting $D = 1.12$ and $\theta = 2.2$ left a good compromise. There were too many spikes at low clamped input (300mV had 14 model and 11 real, and 600mV had 40 model and 32 real), but most of the stochastic spikes (183 of 200) were present.

Fatigue had a minor effect on the results. Removing fatigue did increase the false positives with 900mV input. The original parameters correctly produced all 49 spikes, but added two incorrectly. Removing fatigue added another seven incorrect spikes. It also added an extra eight spikes to the stochastic input. The 300mV and 600mV clamped input remained unchanged. The effect of fatigue was negligible because, outside the clamped 900mV period, there was not a sustained period of high input to cause fatigue to accumulate.

Of the 288 spikes emitted by the model, 26 alignments had two model spikes aligned with a biological spike. The first of these was taken for a timing comparison. Of the 260 directly aligned spikes, the average variance between the model and biological time was 16.3 ms.

The simulation is open to the criticism of testing on the training set. While there are only four parameters, this is still a valid criticism. However, when the model was compared to a second run of the same input on the biological data, the model's fit improved a small amount. In the second run, the biological data produced an extra spike for each of the three clamped input regimes, and one less spike for the stochastic input. Of the 288 spikes emitted, 261 aligned with an average variance of 15.3 seconds.

5 Other Neural Models

The FLIF model presented in this paper is a relatively simple point model, where the model does not consider any spatial components of the neuron. Compartmental models [8, 10] are appreciably more complex, mapping the structure of the entire neuron body. These models can be further refined to include, synaptic delays, ion transfer, and so forth. The FLIF model was compared to the real spike data. Compartmental models can compare, relatively accurately, at the actual voltage level. While they are more accurate, compartmental models are appreciably more expensive computationally to simulate.

A primary motive of the FLIF model is computational efficiency, so that thousands, millions and even billions of neurons can be simulated in real time on relatively simple machines so that the cognitive aspects of behaving agents can emerge from simulated neurons. The trade-off between biological accuracy and computational efficiency is skewed more toward efficiency in the FLIF (and other point) model and toward accuracy in compartmental models.

Simple IF models [11, 27] would not have fared well simulating this biological data. If no activity were retained between steps, it would not have been able to spike with small inputs, or it would have emitted thousands of spikes if the threshold was set low enough to enable spikes with small inputs. If it had retained all activity between inputs, it would have spiked too frequently with small inputs.

The LIF models [3] would have fared much better. As noted in section 4, fatigue only had a significant impact when there was a sustained rate of high input (e.g. two seconds at 900mV).

Another issue that is relevant to both compartmental and firing models is time. It would be relatively simple to modify the FLIF model to have a finer time grain, e.g. .1 ms. per cycle, but that would increase the time needed to simulate each neuron.

The Spike Response Model [9, 19] is another model combining thresholding, refractory periods and randomness. As the spike data accounted for seems, at best, weakly effected by refractory periods, it is likely that this model would not perform particularly well on this data set.

There are of course other higher level models of neurons. For example, a model of cell assembly behaviour [20] models the behaviour of sets of neurons. Similarly, there is a theoretical mapping between adaptive resonance theory [6] and group of neuron behaviour. Clearly, these models could not account for the spike data.

6 Extended FLIF Model

The above FLIF model frequently underestimates the first spike after a period of inactivity. One possibility to account for this, and improve the model is to modify the model so that fatigue could reduce the firing threshold ($F < 0$) in addition to raising it.

With this in mind, a series of modifications to the fatigue rule were implemented and tested. The first was simply to remove the $F < 0$ constraint from the fatigue equation (Eq. 5) leaving Eq. 6.. This however led to rapid firing with no input when $(F * -1) > \theta$. With no input, fatigue will descend until fatigue alone causes the neuron to fire. Fatigue will then increase, but only by a small amount, F_c, and will fire again in a few cycles when fatigue alone again causes the neuron to fire.

$$F_i^{t+1} = F_i^t - F_r \tag{6}$$

Firing with no input is spontaneous activation, and in the biological data described above, there was no spontaneous activation.

With the simple test the simulated neuron spikes at 3.12 seconds. This is the model's version of spontaneous firing. With $\theta = 2.2$ and $F_r = 0.01$, one would expect the neuron to fire at 2.2 seconds, but this is during the inhibitory input phase. Consequently, only after inhibition has ceased, and some of that inhibition has leaked away, does the neuron fire. It then continues to fire as the inhibition is removed, and the fatigue grows due to firing. After 22 steps of firing the model fires every 4 or 5 cycles. In total there are 692 spikes, so the simple substitution of Eq. 5 with Eq. 6 is by itself insufficient.

The simple removal of the $F < 0$ constraint can be improved by resetting fatigue when a neuron fires when the fatigue value is very negative. If fatigue is set to 0 when negative and the neuron fires, the performance using the new fatigue decrementing rule (Eq. 6) is better than without, and is now comparable with the original fatigue rules.

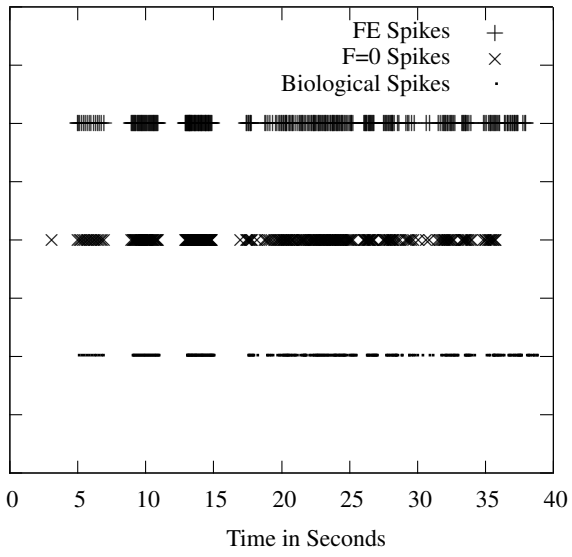

Figure 4. Spikes produced by the biological neurons, the fatigue reset to 0 model, and the exponential fatigue reduction model over 38 seconds

Figure 4 shows firing behaviour over time. The biological data shows the initial lack of spikes, and then the lack of spikes when there is no input. The F=0 data shows the simulation with fatigue reset to 0 after firing with fatigue negative. Note the first spontaneous spike just after three seconds. The FE data shows the spikes from the exponential fatigue recovery rule described below; note that the spontaneous spike no longer occurs.

When fatigue was reset to 0 after a neuron fired due to fatigue, there were 322 spikes. Of the 322 spikes emitted by the model, 42 alignments had two model spikes aligned with a biological spike. The best of these was taken for a timing comparison. Of the 275 directly aligned spikes, the average variance between the model and biological time was 14.7 ms.

This still had the problem of a spike at 3.12 seconds. While neurons fire spontaneously, the biological neuron being modelled does not spontaneously fire under the input regime being explored. So, the model was modified so that it recovered from fatigue less when fatigue was negative (Eq. 7).

$$\begin{aligned} F_i^t >= 0 &\rightarrow F_i^{t+1} = F_i^t - F_r \\ F_i^t < 0 &F_i^{t+1} = F_i^t - (F_r/k) \end{aligned} \tag{7}$$

The first test that was run was with $k = 2$. This meant that fatigue reduced slower, but there was still a spontaneous spike at 4.39 seconds. As there are no spontaneous spikes in the data, it only needs to spike after the first positive input at 5.0 seconds. Setting $k = 3$ did this.

Fatigue was reset to 0 after a neuron fired due to fatigue, but fatigue reduction slowed (by $k = 3$) when fatigue was negative. There were 295 spikes with 28 alignments where two model spikes aligned with a biological spike. Of the 267 directly aligned spikes, the average variance between the model and biological time was 16.3 ms.

A further extension was to modify the way negative fatigue was changed after firing. Instead of setting fatigue to 0, it could be changed to F/m. This would have the advantage of allowing under fired neurons to fire rapidly more than once. It required a change when fatigue approached 0 from a negative value, or fatigue would always remain negative, so the regime changed at -.25. The new fatigue increase rule is Eq. 8.

$$\begin{aligned} F_i^t >= -.25 &\rightarrow F_i^{t+1} = F_i^t + F_c \\ F_i^t < -.25 &\rightarrow F_i^{t+1} = F_i^t / m \end{aligned} \tag{8}$$

When $m = 4$, and $k = 3$ the system fired 297 spikes. 266 of them aligned correctly (and 30 pairs aligned), with an average difference of 15.6 ms. Values of $m = 2$ and $m = 8$ performed slightly worse.

Run	Neurons Fired	Double Aligned	Correctly Aligned	Time Difference
Real	294	0	294	2.5
Standard	288	26	260	16.3
F0	322	42	275	14.7
F3	295	28	267	16.3
F3-4	297	30	266	15.6
FE	299	21	277	16.9

Table 1. Relation between biological and model neural firing

A final variant of the fatigue rule was explored. Instead of reducing the fatigue by a factor of the fatigue when it was negative, an exponential function was used (Eq. 9).

$$F_i^t < 0 F_i^{t+1} = F_i^t - (-4)^{(3 - F_i^t)} \tag{9}$$

This was motivated by having the change being roughly equal to $F_r = .01$ at 0 for a smooth function. Additionally, it needed to prevent the neuron spiking without input in the first 5 seconds, but eventually enable it to spike. This function managed both of these constraints. With this new fatigue rule, setting $F = F/2$ when the neuron fires due to fatigue, and $D = 1.16$, the model with probably the best fit of those simulated was developed. This model fired 299 times, only 21 of the biological spikes had two simulated spikes aligned to them. 277 spikes were correctly aligned with an average time variance of 16.9 ms.

Table 1 presents the results of the simulations. The first row represents the biological data, and the other rows represent the runs reported above. Note that the best possible model would have erred

by 2.5ms due to the 10ms simulated cycle speed. The Standard row refers to the simulation run with the standard fatigue model, where fatigue is always 0 or greater. The other models allow fatigue to become negative, and thus to cause the neuron to fire with no input, or to fire faster with less input. The rows differ by the fatigue change when fatigue is less than 0. The F0 row is the standard model but resets fatigue to zero $F = 0$; F3 reduces fatigue by $F_r/3$ when fatigue is negative; and the F3-4 row reduces fatigue by $F_r/3$, and sets it to $F/4$ when the neuron fires and fatigue is less than -.25. FE refers to the system with the exponential fatigue reduction rule when fatigue is negative; this also resets fatigue to $F/2$ when the neuron fires and fatigue is less than -.25, and has $D = 1.16$.

None of these models is clearly superior to the other models, though the exponential model might claim to be the best; however, it does have poorer time difference. Resetting fatigue to 0 creates more spikes and more aligned spikes, but has more doubly aligned spikes.

Again, after model and parameter development, all of the above model were tested on the second set of biological data. The performance of all models improved slightly on this data.

Reducing the fatigue rate when stored fatigue is negative produces the right number of spikes, and shows no spontaneous firing. In the biological data, no spontaneous firing occurs. Still, in many cases, without input, neurons spike spontaneously [21]. Spontaneous activation fits in nicely with another type of model, the Boltzmann machine [2]. In the Boltzmann machine, without activation, the neuron fires at regular intervals. Increased activation makes the neuron fire earlier. To some extent, the revised FLIF model incorporates this behaviour.

This spontaneous firing is important for a range of reasons [16]. Perhaps the chief reason is that without spontaneous activation, when using a strictly Hebbian learning rule, firing cannot easily move beyond the neurons that are directly activated by the environment. This is a modelling problem, but there is biological evidence that spontaneous firing is needed to properly develop vision [30]. In simulated systems, since no neuron that is not directly stimulated receives activation, they will not fire. Since they do not fire, the Hebbian learning rule will not increase the weight from firing neurons to them. With spontaneous firing, those neurons that are not directly stimulated by the environment can co-fire with those that are, and then the synaptic weights can increase. The biological evidence on visual development supports this line of argument from simulation.

7 Conclusion

This paper has described a FLIF neural model, discussed some biological neural data, and derived parameters for the FLIF model from that neural data. Thus, the FLIF model with parameters set to $\theta = 2.2$, $D = 1.12$, $F_c = 0.045$, and $F_r = 0.01$ is a relatively faithful model of this particular rat neuron.

The exponential fatigue recovery rule performed even better getting 277/294 (94.2%) spikes correct. Of course, this fatigue rule is more computationally expensive to run. This final phase of testing was relatively brief, and further modifications could lead to further improvements in model fit. Note that in all of the reported results, accuracy averaged well within two cycles.

The paper has indicated that fatigue can play a useful component of a neural model. This is the case when there is a significant amount of input for a reasonably long duration. Fatigue may also be used in learning to account for particular psychological phenomena [22].

Further modification of the fatigue rule has shown similar fits to the biological data. Thus, the new models can fire spontaneously, while still fitting the data.

Moreover, further biological data should support better parameter fitting and neural model development. In particular, data on spontaneous firing should support a more biologically accurate fatigue model.

It should be noted that there has also been research in developing systems to automatically fit neural models to biological data [28] including the data used in this paper. It is useful to note that integrate and fire models with adaptation (fatigue) have been relatively successful.

It should also be emphasised that this is a model of a particular neuron. It is almost certain that, even when applicable, the FLIF model of different neurons would require different parameter settings. This paper has only considered one model neuron, though it was the first the authors actually tried to model.

None the less, the four free parameters of the model have been set so that they account for 90% of the spikes relatively well. Similarly, several variants of the fatigue model produce similar and perhaps superior results. Consequently, it seems that these models are of reasonable biological fidelity.

Acknowledgements: Thanks to Middlesex University's School of Engineering and Information Sciences for partial funding of this work.

REFERENCES

[1] L. Abbott, 'Lapicque's introduction of the integrate-and-fire model neuron (1907)', *Brain Research*, **50**, 303–304, (1999).

[2] D. Ackley, G. Hinton, and T. Sejnowski, 'A learning algorithm for boltzmann machines', *Cognitive Science*, **9**, 147–169, (1985).

[3] D. Amit, *Modelling Brain Function: The world of attractor neural networks*, Cambridge University Press, 1989.

[4] R. Belavkin and C. Huyck, 'Conflict resolution and learning probability matching in a neural cell-assembly architecture', *Journal of Cognitive Systems Research*, (accepted).

[5] T. Berger and R. Naud. Quantitative single-neuron modelling, 2009.

[6] G. Carpenter and S. Grossberg, 'The art of adaptive pattern recognition by a self-organizing neural network', *IEEE Computer*, **21**, 77–88, (1988).

[7] M. Chacron, K. Pakdaman, and A. Longtin, 'Interspike interval correlations, memory, adaptation, and refractoriness in a leaky integrate-and-fire model with threshold fatigue', *Neural Computation*, **15**, 253–278, (2003).

[8] P. Dayan and L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, 2005.

[9] W. Gerstner and J. vanHemmen, 'Associative memory in a network of spiking neurons', *Network*, **3**, 139–164, (1992).

[10] A. Hodgkin and A. Huxley, 'A quantitative description of membrane current and its application to conduction and excitation in nerve', *J. of Physiology*, **117**, 500–544, (1952).

[11] J. Hopfield, 'Neural nets and physical systems with emergent collective computational abilities', *Proc. of the Nat. Academy of Sciences USA*, **79**, 2554–8, (1982).

[12] C. Huyck, 'Cell assemblies as an intermediate level model of cognition', in *Emerging Neural Architectures based on Neuroscience*, eds., S. Wermter, J. Austin, and D. Willshaw, 383–397, Springer, (2001).

[13] C. Huyck, 'Creating hierarchical categories using cell assemblies', *Connection Science*, **19:1**, 1–24, (2007).

[14] C. Huyck, 'CABot1: a videogame agent implemented in flif neurons', in *IEEE Systems, Man and Cybernetics*, (2008).

[15] C. Huyck, 'A psycholinguistic model of natural language parsing implemented in simulated neurons', *Cognitive Neurodynamics*, (2009).

[16] C. Huyck and R. Bowles, 'Spontaneous neural firing in biological and artificial neural systems', *Journal of Cognitive Systems*, **6:1**, 31–40, (2004).

[17] C. Huyck and H. Ghalib, 'A neuropsychological framework for advancing artificial intelligence', in *AAAI fall symposium on Biologically Inspired Cognitive Architectures*, p. 87, (2008).

[18] R. Jolivet, R. Kobayashi, A. Rauch, R. Naud, S. Shinomoto, and W. Gerstner, 'A benchmark test for a quantitative assessment of simple neuron models', *Journal of Neuroscience Methods*, **169**, 417–424, (2008).
[19] R. Jolivet, T. Lewis, and W. Gerstner, 'The spike response model: A framework to predict neuronal spike trains', *Springer Lecture Notes in Computer Science*, **2714**, 846–853, (2003).
[20] S. Kaplan, M. Sontag, and E. Chown, 'Tracing recurrent activity in cognitive elements(trace): A model of temporal dynamics in a cell assembly', *Connection Science*, **3**, 179–206, (1991).
[21] L. Katz and C. Shatz, 'Synaptic activity and the construction of cortical circuits', *Nature*, **274**, 1133–1138, (1996).
[22] L. Kleinsmith and S. Kaplan, 'Paired-associate learning as a function of arousal and interpolated interval', *Journal of Experimental Psychology*, **65:2**, 190–193, (1963).
[23] T. Kohonen, *Self-Organizing Maps*, Springer, 1997.
[24] P. Lansky, P. Sanda, and J. He, 'The parameters of the stochastic leaky integrate-and-fire neuronal model', *Journal of Computational Neuroscience*, **21**, 211–223, (2006).
[25] W. Maas and C. Bishop, *Pulsed Neural Networks*, MIT Press Cambridge, MA, 2001.
[26] H. Markram, 'The blue brain project', *Nature Reviews Neuroscience*, **7**, 153–160, (2006).
[27] W. McCulloch and W. Pitts, 'A logical calculus of ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics*, **5**, 115–133, (1943).
[28] C. Rossant, D. Goodman, J. Platkiewicz, and R. Brette, 'Automatic fitting of spiking neuron models to electrophysiological recordings', *Frontiers in Neuroinformatics*, **4:2**, (2010).
[29] D. Rumelhart and J. McClelland, *Parallel Distributed Processing*, MIT Press, 1986.
[30] W. Singer, 'Development and plasticity of cortical processing architectures', *Science*, **270**, 758–764, (1995).